MASTER IN COMPUTER SCIENCE

Topic extraction in words networks

Bertrand, Florian

Award date:
2017

Awarding institution:
University of Namur

Abstract

This Master Thesis explores an original solution to analyze text document collections. This method is constructed with methods from Data Mining, Text Analytics and Social Network Analysis. The basic idea underlying it is to create a network in which words are entities and relationships represent how often these words are used together, with the aim to find topics as words communities. After a concept definition part and an analysis of the benchmark dataset with existing techniques, the development of the method is explained. First, the parsing of the document is presented. Then, the method of creation and selection of the links, which is based on frequent item set mining, is exposed. Finally, the results of the analysis with both visualization and community detection algorithms are compared with results obtained using existing techniques.

I would like to thank everyone who helped me through this project and believed in it. Especially:

My professors, especially my promoters Pr. Benoît Frénay and Pr. Renaud Lambiotte.

Each employee of SAS® Institute who helped me by sharing their knowledge and answering my many questions. Especially my reader Joline Jammaers, Véronique Van Vasselaer and Stéphane Marchand with who I am working for many years now.

Each proofreader: my friend Patrick Foissac, my parents and my brother Yannis.

My girlfriend Elodie who has been supporting me for so many years.

Contents

List of Figures

Introduction

The objective of this thesis is to develop and explore an original method to explore documents collection. The basic idea underlying it is to create a network in which words are entities, and relationships represent how often the words are used together. Two words are "friends" if they often appear in the same text, which can be either a document, a paragraph or a sentence. A community of words would then represent a bag-of-words which are often used together, so in other words a theme. The first expected advantage of such a method would be its visualization and its ability to be explored interactively. The second foreseen advantage would be the ability of the method to recognize topics of different sizes at the same time.

The research question behind this objective is: "is it possible to automatically build a network of words in which communities would represent different topics? If yes, how? And how does it perform?". So first, the thesis aims to determine if it is possible to create such a network. Then, it aims to find a way to build it: what are the different techniques and algorithms used in the different steps and in which order? Finally, it aims to compare the performance of such a method with a regular method. These questions are answered together in the following chapters.

To develop this method, a software package had to be chosen. The best package would present the following main characteristics: to be fully customizable to fit such an original method, to have some standard processes in the area of study to be able to easily validate the first steps and finally to share the same data format between the different components of the architecture (eg metadata). That is the reason why SAS was chosen. On one hand, commercial software packages with a drag and drop interface, such as SPSS or Knime, are not well suited for such a customization, while on the other hand, programming languages used in analytics such as R, Python or Matlab do not have standard processes ready to be used by simply dragging them in

a process flow, nor a common data format between components that allows to easily connect them.

This thesis is divided into two main parts: the first one focus on the theoretical concepts underlying the developed method and the second one on the application of these concepts to develop a consistent method. The first part is divided into three chapters representing the three main topics of advanced analytics which are used in the method: the Data Mining, the Text Analytics (and Text Mining) and the Social Network Analysis. The second part is then split into four chapters. The first one (chapter number four) shows the results of a common Text Mining analysis applied to the document collection under study in order to have a basis for comparison with results from the developed method. Then, the following chapter presents how Text Analytics techniques are used to parse the document, to select useful words as input and to transform the data in order to be processed further. After that, chapter six presents how data mining is applied to select meaningful relationships (words that co-occur more often than they occur alone). Finally, chapter seven shows both a visual analysis of the network and an analysis of the resulting communities from different community detection algorithms.

Part I

Theoretical explanations

Chapter 1

Data Mining

We must first define Data Mining before we can define and explain what is Text Mining, in order to better understand the link between the two domains. Friedman (1998) tried to find a definition with Data Mining characteristics and its differences with Statistics. So it could be defined as follows: Data Mining is the process of automatically extracting knowledge from huge volumes of structured data in order to support business decisions.

There are two different domains inside Data Mining:

- Exploratory (unsupervised learning): Clustering, Association Analysis, Link Analysis... Techniques which make trends, patterns or groups emerge from data.
- Predictive (supervised learning): predicts an unknown target variable with means of other variables values.

The main difference between them is that predictive analysis requires a target to build a model, while the other does not.

1.1 Data Mining Methodology

As stated in the Data Mining definition, it is a process that is composed of different steps. There are some methodologies for the Data Mining process as explained in Azevedo (2008). The three main methodologies which are standards in the industry are KDD (Knowledge Discovery in Database), SEMMA (Sample, Explore, Modify,

Model, Assess) and CRISP-DM (Cross-Industry Standard Process for Data Mining). In fact, these methodology can be summarized by SEMMA, since the most difference lies in the cyclical fashion of the others and the business understanding phase which is missing in SEMMA.

1. Sample means separating data with a trade off between having enough data to extract significant information, but not too much data, to avoid time consuming processing. It is also the step during which we sample the dataset into a training set, a validation set and sometimes a test set (this is done for classification purpose).

2. Explore means searching for "anticipated relationships, unanticipated trends or anomalies" (eg outliers). The aim is to better understand the data but also to get new insights on them.

3. Modify is a broad concept: it could mean to impute missing values, to normalize data by applying transformation functions (such as log or square root), to clean data (eg from noise or outliers), to consolidate variables or to apply data reduction methods such as Principal Component Analysis (PCA) or Discriminant Analysis.

4. Model is the step in which a model is applied: a decision tree, a neural network, a regression model...

5. Assess is the last step. Here the different models are compared regarding a defined assessment criterion (misclassification rate, average squared error, profit maximization...). After this step the model is ready to be deployed.

1.2 Data Exploration

Exploratory analysis is summary statistics and unsupervised learning, that means it does not require having a target variable, though some techniques can use it. This kind of analysis is descriptive rather than predictive.

1.2.1 Association and Sequence Analysis

This section focuses on Association Analysis and gives a brief overview of sequence analysis. Association analysis is also known as frequent item set mining or pattern set mining as stated in Hossain *et al.* (2014).

Association Analysis seeks which items are bought by the same client. It does not mean items should be bought together, nor in a particular sequence, simply that someone who buys item A is likely to buy item B ($A \Rightarrow B$). To better understand the following chapter it is important to get a closer look to the rules characteristics. P(A) means the probability of finding A in a transaction (so the event is "A is in the trolley").

The first characteristic of a rule is its support. It represents the percentage of all transactions in which the rule occurs. For the $A \Rightarrow B$ rule, $Support = P(A, B)$. Support is associative: support of $A \Rightarrow B$ is equal to support of $B \Rightarrow A$.

The second one is the confidence of a rule. It represents the conditional probability to have the right hand side of the rule given the left hand side. For the $A \Rightarrow B$ rule, $confidence = P(B|A)$.

The third one is the expected confidence. It represents the percentage of all transactions in which the right hand side occurs. So it represents the probability to get the right hand side without any condition. For the $A \Rightarrow B$ rule, $expectedconfidence = P(B)$.

Finally, the last one is the lift. The lift is computed as the confidence divided by the expected confidence. For the $A \Rightarrow B$ rule, $lift = confidence/expectedconfidence$. It measures the strength of the association: given the left hand side, how much time is likelier to find the right hand side than randomly. So, as an example, a lift of 2 means that given the left hand side, it is two times likelier to find the right hand side than randomly. The lift is associative: $A \Rightarrow B$ lift is equal to $B \Rightarrow A$ lift.

$$P(B|A)/P(B) \stackrel{?}{=} P(A|B)/P(A)$$

$$P(B|A) * P(A) \stackrel{?}{=} P(A|B) * P(B)$$

$$P(A, B)/P(A) * P(A) \stackrel{?}{=} P(A, B)/P(B) * P(B)$$

$$P(A, B) = P(A, B)$$

Saving Account \ Checking Account	No	Yes	
No	500	3,500	4,000
Yes	1,000	5,000	6,000
	1,500	8,500	10,000

Table 1.1: Example from banking domain used to illustrate computations

Support	=	5,000 / 10,000	=	50%
Confidence	=	5,000 / (1,000 + 5,000)	=	83%
Expected Confidence	=	(3,500+5,000) / 10,000	=	85%
Lift	=	0.83/0.85	=	0.97%

Table 1.2: Example of Support, Confidence and Lift computation for rule $SavingAccount \Rightarrow Checkingaccount$

Concrete example of the characteristics computation is given in array 1.1 and array 1.2. Array 1.1 contains counts of people having a checking and/or saving account or not at all. Array 1.2 present the computation of the different rules characteristics given the numbers of array 1.1. Another more visual example is given in figure 1.1[1]. In this picture, the trolleys represent the transactions, and the colored box represent the items. Support represents the percentage of transactions in which both elements are. Confidence represents the percentage of transaction containing A where B is also present.

Sequence Analysis finds how buying an item increases odds to buy another one in the future. It is a special case of association analysis with a time notion. A good example: someone who bought a TV is likelier to buy a home theater than someone who did not buy a TV before.

1.2.2 Clustering

The aim of clustering is to group subsets of data which are close to each other. It is at the same time a maximization and minimization process: minimization of variance (differences) within the same clusters and maximization of variance between different clusters. It means the population inside a cluster should be quite homogeneous, but observations from two different clusters should be different enough.

[1]Image taken from SAS Institute (2011)

Rule	Support	Confidence
A ⇒ D	2/5	2/3
C ⇒ A	2/5	2/4
A ⇒ C	2/5	2/3
B & C ⇒ D	1/5	1/3

Figure 1.1: Association Analysis trolleys example

There are two main kinds of clustering algorithms: hierarchical or agglomerative. The second one requires to have an idea about the number of needed clusters, while the first one builds clusters as a partition of a bigger one. Then the user has to choose the number of clusters which seems best suited regarding his study.

1.3 Data Preparation

The data preparation phase depends on the previous and the next steps: the previous one because it shows which transformation(s) should be made to the data, and the next one because the modeling method also determines which transformations should be applied (Baesens *et al.* (2015)).

1.3.1 Data Transformation

Since there are plenty of possible transformations, only a few, the most commonly used, are discussed.

For example, imputing missing values is done if the exploratory phase showed missing values and if the modeling method is not able to handle missing values very well (eg Regressions or ANN).

Another example: data normalization is only applied to data with a non-normal shape (which is seen during the exploratory phase) and if the modeling method requires normal data (eg Regressions).

1.3.2 Dimension Reduction

This step is essentially done with high dimensional datasets. The most well-known is PCA (Principal Component Analysis). However, there are other useful methods: variable selection methods, Discriminant Analysis, Clustering... and even models can be used such as Decision Trees or Regressions. Decision Trees can group categorical variables in new variables with less categories, while Regression, with a variables selection method, can also be used to select variables. Such a selection method builds models based on different subsets of variables, the subset used to build the best model is then chosen.

1.4 Modeling

Modeling means (in the chosen methodology) predictive analysis: it outputs models which can be deployed outside the Data Mining software to score new data. It requires to have a target on which the model can be built. Nevertheless, the target can be constructed during the previous steps (eg a cluster). This target can be continuous, ordinal or categorical. The aim of a prediction can be estimation, ranking or decision (Baesens *et al.* (2015)). Furthermore, in order to build accurate models, the data has to be sampled in a training set, a validation set and sometimes a test set, to avoid overfitting.

Decision trees are sets of rules which are applied sequentially to segment data into subsets to predict the outcome of a target variable. At each split of the tree, the idea is to divide data into subsets where cases are quite homogeneous regarding the target variable. They are represented as trees where each split represents a rule and branches the possible outcomes of the rule. Their main advantage is their ease of understanding and deployment, but also that they are able to cope with missing

data.

1.5 Model Assessment

The different models built need to be assessed to select the best one, regarding a user's defined criterion: profit maximization, cost minimization, average squared error, misclassification rate.... It depends mainly on the aim of the prediction.

1.6 Model Deployment

Although deployment means applying the built model on new data, in general terms it means using the found knowledge. So for predictive modeling, it means scoring new data, and in some cases exporting the model outside of the Data Mining software (in C, Java...) to build ones' own prediction application or include this prediction in an existing application. But with exploratory analysis it could mean, for example, to use the associations found in the association analysis: increasing the diapers price and decreasing the beer price to improve profit. In clustering, it could mean exporting clusters to classify new observations into clusters.

Chapter 2

Text Analytics

With the exponential growth of what we call Big Data, the utilization of new technologies able to cope with this unstructured data emerged and an area of those techniques is Text Analytics. As stated in Zanasi (2007), textual data represents about 17% of data mined by companies while according to Gartner Group, cited in Ananyan & Kiselev (2001), the textual data would represent about 90% of the organizations available data. Text Analytics is the domain which copes with this textual data, by trying to extract knowledge from this huge amount of information. Text Analytics can be divided in two approaches: a top-down approach and a bottom-up approach. The first one requires previous business knowledge, it is the approach of content categorization (it can also be combined with the bottom-up process). It does not use a corpus of documents in opposition to the other approach, it simply checks if a document fulfills some criteria, previously defined by a Business Expert. Information Extraction is also a top-down approach: a Business Expert must define the entities to extract through regular expressions or ontology.

According to Feldman & Sanger (2007), Text Mining with supervised learning is also a top-down approach since it requires some insights into the data: the target or response variable on which the classifier is built. The other approach, bottom-up, extracts knowledge from a corpus of text, without prior knowledge of what these texts are about. That is where we find Information Retrieval and Text Mining (unsupervised), but Information Retrieval could also be classified as a bottom-up through the automatic summarization. Figure 2.1 shows the different areas from Text Analytics.

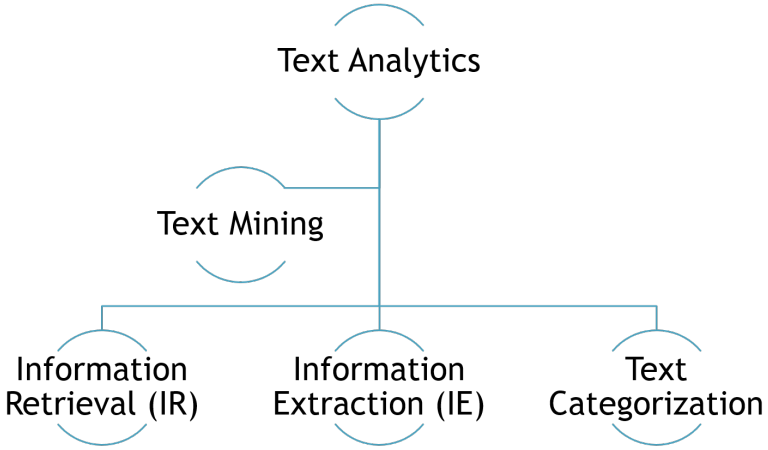


Figure 2.1: Text Analytics areas

Many papers or books use the term Text Mining synonymously with Text Analytics, but there is a shade between both domains: Text Mining is a subarea of Text Analytics. Text Analytics encompasses Text Mining, Information Retrieval (IR), Information Extraction (IE), Content Categorization... as stated in SAS Institute (2014). Other definitions can be found, Zanasi (2007) gives almost the same definition but for Text Mining. So according to the author, Text Mining encompasses techniques such as Machine Learning, Data Mining, Linguistic... Nevertheless, it should be better to use the same terms for texts as for data: Data Mining is a subarea of Analytics. That is also, broadly speaking, the definition given by Ananyan & Kiselev (2001), Chen (2001) and Gupta & Lehal (2009).

Text Mining could be defined as the process of making knowledge from textual data. While Data Mining uses only structured data, Text Mining is able to cope with semi-structured (eg xml) and unstructured data (text documents in general). It includes similar analytical procedures as Data Mining: pattern discovery/exploratory analyses and predictive modeling. SAS Institute (2014) considers Text Mining to have the following characteristics: it begins with a corpus of texts, it creates a dictionary regarding terms in the corpus, it provides many metrics to gauge document contents, it "derives structured vector(s) of measurements for each documents" from the corpus and finally it applies analytical methods with this structured vector as input.

2.1 Text Analytics Areas

Here are described Text Analytics areas without Text Mining which is explained in details further in this text.

2.1.1 Information Extraction

Information Extraction (IE) is the process of extracting knowledge from text(s), mainly entities (such as named entities: localizations, names, organizations, telephone numbers...) which fill predefined forms (Pazienza (2007)). Another Information Extraction method is Texts Summarization. Entities Extraction is generally problem-dependent as stated in Feldman & Sanger (2007). Custom entities can be created through two techniques:

- Dictionary/ontology: for entities with a limited number of values (eg cities or surnames).
- Regular Expressions: for entities which follow a "pattern" (eg email: an email address is always built with extended alphanumeric characters, followed by @, followed by alphanumeric characters, a period and a domain name).

2.2 Natural Language Processing

When we already have a "Corpus" of texts, we are able to build the term-document matrix: the way texts are represented numerically. This matrix has terms as documents and documents as features. It is one of the roles of Natural Language Processing (NLP) which includes tokenization, stemming, POS tagging, parsing...

2.2.1 Parsing

First, to understand how this process works, let us define what tokens and terms are. Tokens are strings extracted from a text which are separated by separators (spaces, punctuations and so on), while a term is a set of (following) strings or substrings which have a particular meaning. Let us take examples to illustrate this (which come from Knappen (2015)).

"White House" is processed as two tokens, "White" and "House", because there is a separator (here a space). Although we have two tokens, there is a meaning to

the expression "White House" which is the house of the American President, so it is a term. Another example is "washingmachine": we have one token since there is no separator, though it is the combination of two terms: washing and machine.

Moreover, phrasal verbs in English have their particular meaning if and only if the verb and the preposition are read together.

2.2.2 Stemming

The next step to build the term-document matrix is to stem words. Stemming is the process of regrouping nouns in their singular form and verbs in their infinitive. While "race" and "races" are two different tokens, term groups all occurrences of these tokens in one term: "race". Once again, while "go" and "went" are two different tokens, they are grouped in one term: go. To translate tokens into terms, the parsing engine needs a dictionary. That is why it is so important to make the distinction between terms and tokens: without a dictionary in the analyzed language, we are only able to make analyses based on the token-document matrix.

As described in Zhao (2013), stemming algorithms stem words in their radical form, so they make no distinction between verbal form and noun. As an example, "update" could be seen as a verb or a noun. A stemming algorithm stems them in the same term "update".

2.2.3 Start/stop words and POS tagging

"Start words" and "stop words" are lists of words, specified by the user or downloaded from the Internet, which specify the only words we are interested in ("start words") or the excluded words ("stop words"). While "start words" are most of the time built from technical or business dictionaries, to get only the terms that correspond to the studied topic, "stop words" aim to remove noise from the text. As briefly explained in SAS Institute (2014), these "stop words" are those which provide no descriptive nor predictive value, and so only create noise in the data.

In addition to these start/stop words, it is possible to reject complete "grammatical entities" (POS, Part-Of-Speech) such as determiners, conjunctions, prepositions, pronouns and so on. For example, determiners have no explicative power: what could "the" bring in a model besides noise? These steps are very important because they restrict considerably the term-document matrix by avoiding "meaningless" words,

which makes further analysis less time-consuming.

2.2.4 Other possible techniques available during NLP

Information Extraction can be carried out, especially Concept Extraction (eg n-gram) and named Entities Extractions as already explained. N-grams are n words in a row which often appear together. Most of the time we only seek bi-grams because a n-gram search (with a higher n) is time-consuming.

Some Text Mining software have also spell check tools which correct them. There is a trade off regarding the processing time because spell checks require lots of time. But if they are not corrected, the term-document matrix is bigger (regarding the amount of misspellings found in the corpus) and moreover, the explicative power of some words is lost. Let us take an example: if in some documents people have written "steal" in place of "steel", the explicative power is shared by both words, while steel should have all the power.

Searching for synonyms has the same aim as checking out typos: if the same idea is spread into two different words, the explicative power of this idea is split between the two terms. As an example, if someone uses "automobile" and "car" to avoid repeating always the same word, there are two terms, which are synonyms and have the same meaning. The issue is that these terms are less represented than if there was one unique word. In our example, if the writer is speaking about buying a "blue car" or a "blue automobile", the results of the analysis could link only "blue" to this person.

2.2.5 Assigned weights

The values inside the term-docs matrix correspond firstly to the frequency a term is encountered in a document. The frequency weight is often called in the literature "local weight" and it refers only to weight between a document and a term (Ibekwe-SanJuan (2007)). Regarding the aims of the analysis, rather than a simple frequency count, it is possible to assign the logarithm of this frequency, a binary variable which indicates only if a term is present in a document or not and so on.

Beside local weights, there are "global weights" which are built regarding the entire frequency of terms, which is the reason why they are essentially called "term weights" (Ibekwe-SanJuan (2007)). There are some methods to compute them,

among which the entropy criteria, the Inverse Document Frequency and the Mutual Information. The name entropy is a bit misused because the formula found in SAS Institute (2014) differs from the formula used in Data Mining for decision tree building.

$$G_i = 1 + \sum_{j=1}^{d_i} \frac{p_{i,j} log_2(p_{i,j})}{log_2(n)}$$

where G_i is the entropy for word i, d_i are the documents where word i appears, $p_{i,j}$ is the frequency of the word i in document j divided by the frequency of the word i in the document collection, and n is the total number of documents. Given the entropy of Shannon (2001), the text mining entropy can be defined as 1-normalized entropy (normalized due to the division by $log_2(n)$. This formula ranges from 0 to 1, 0 meaning no information at all, while 1 means high information. As an example, a word that occurs one time in only one document has an entropy of 1, while a word that is one time in every document has an entropy of 0.

The Inverse Document Frequency (IDF) is simpler than the previous one, it simply computes the log of the inverse ratio of the number of documents in which the term i appears regarding the total number of documents.

$$G_i = 1 + log_2(\frac{n}{d_i})$$

The range of the returned value varies from one to ∞. The bigger the value is, the more information the word provides. It is the most common frequency weight shown in literature. It is presented in Ibekwe-SanJuan (2007), Feldman & Sanger (2007), Milié-Frayling (2007)...

However, care must be taken with these two formulas. Indeed, both return extremely high value for words that appears only once in only one document. Such a word is probably not useful because too discriminating. That is the reason why it is so important to filter the rare terms (eg terms which are not in at least x documents).

The last one is the Mutual Information criteria (also known as Information Gain (Feldman & Sanger (2007))). There are many differences with the previous one. First its interpretation is the inverse of the other, so closer to zero means more information. Then, the biggest difference is that Mutual Information requires a target. So it is only used when there is a target variable.

$$G_i = \max_k [log_{10}(\frac{P(t_i, C_k)}{P(t_i)P(C_k)})]$$

Where k is the number of levels of the categorical target variable, t_i is the proportion of documents containing term i and C_k is the proportion of documents having this target level.

2.3 Data Exploration

In the same way as Data Mining processes, Text Mining needs to explore and understand data. It is done with two main tools: Concepts Linkage graphs and Concepts Exploration. There used to be another way of exploring textual data which is less utilized by now: creating maps of words where islands on the ocean represent associated words as shown in Garnier (2007). It is also possible in this step, because the corpus content has been indexed, to query the corpus.

2.3.1 Concept Linking

Concept Linking is an unsupervised learning method. It aims to show how terms are related to others in documents, so it is to show the correlation between the presence of one word and the presence of another. It could be seen as an adaptation suited for text of Association Analysis. As stated in Fan *et al.* (2006) and Gupta & Lehal (2009), the Concept Linking tools are specifically useful for users who are searching for current associations of words. The best example, cited in the previous papers is the association that has been found by Swanson (1988). By querying manually medical reports whose title(s) contained the word "Migraine", he found that the word "magnesium" was often used with the word "Headaches". He therefore hypothesized that there was a relation between a lack of magnesium and suffering from migraines. It is important to notice that there were no prior direct links in the literature between magnesium and headaches. Thanks to his findings, experiments were made and it is now proved there is a link between magnesium and headaches.

Figure 2.2 shows an example of concept linking applied to the movie dataset with the word "chase". The picture shows the association between this word and eight other words, the thickness of a link represents the strength of the association. When placing the mouse over the central word chase, it shows a tooltip with "93/93",

meaning the word is in 93 documents. The tooltip on the word "bond" ("16/44") means that bond is in 44 documents and 16 of those also include the word "chase". The symbol "+" in front of "chase", "bond", "car", "action", "explosion", "car chase" and the other "chase" means it includes also derivations from the word which were stemmed ("chases" transformed to "chase").

The fact that "chase" is twice in the network is explained by the fact that "chase" can be either a noun, either a verb. This difference is made by the POS tagging. However, as one can imagine with this small network, it can be an issue: "chase" and "chase" co-occurred often, but the information carried by this meaning of "chase" is spread into two words, which become two nodes. Another thing to point out is that "car chase" co-occurs frequently with "car" and "chase". Obviously, if the concept "car chase" occurs in a document, both "car" and "chase" also occur. These two problems are discussed more in details in section 5.2.

Concept linking is very close to the objectives of this thesis. However, in opposition to them, concept link as-is does not build an entire network, it builds small networks from a starting point. If it is useful to explore relation between concepts under study, it is impossible to find association between two words that are not under study unless by testing concept linking for every words.

2.3.2 Text Clustering

Text clustering is the same concept as clustering in Data Mining, but the techniques used are different because of term-doc matrix properties (Mandreoli *et al.* (2007)). To be precise, there are two properties which would lead to issues if they were used with standard clustering: the curse of dimensionality (the number of variables) and the sparsity of the matrix.

In general, the most used technique to overcome these drawbacks is the Latent Semantic Indexing (LSI), also known as Latent Semantic Analysis (LSA) or Vector Space Model (VSM) which performs an SVD (Singular Value Decomposition). SVD is a matrix factorization technique used in different fields as a dimension reduction technique. As explained by Albright (2004), the basic idea of SVD is to decompose the term-doc matrix in three matrices (U, Σ, V) so that $A = U\Sigma V^T$ (see figure 2.3[1]). Where A is the original term-doc matrix (of size $terms \times docs$) and U and V are two

[1]Image taken from Albright (2004)

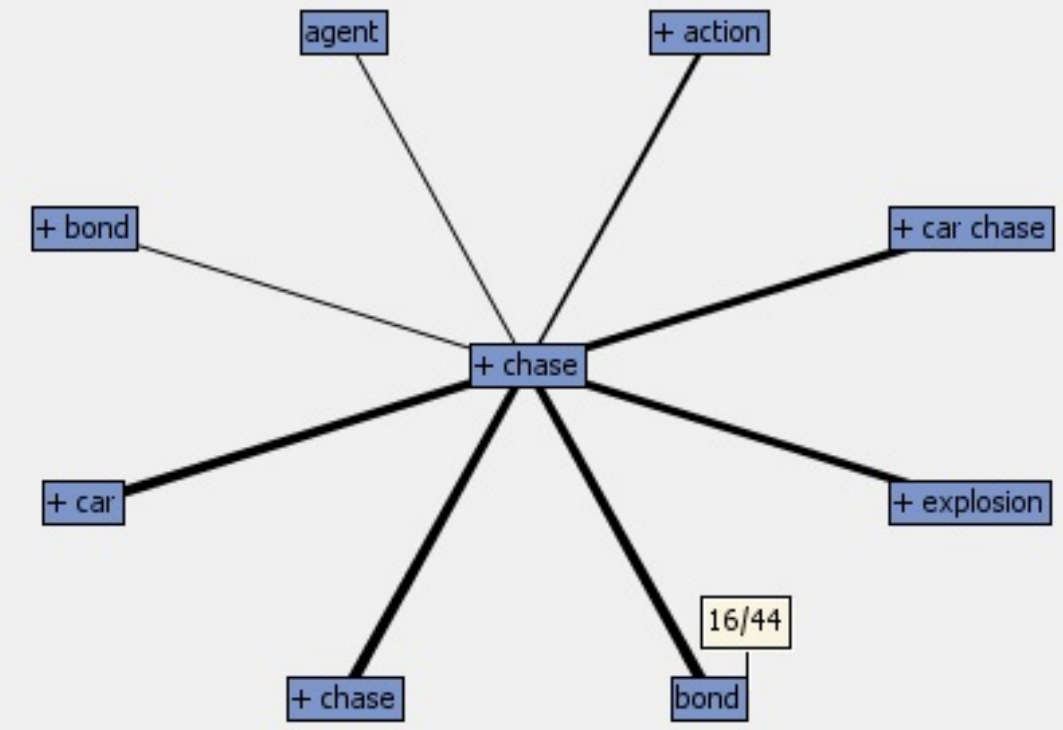


Figure 2.2: Example of concepts linking from the movie dataset with the word "chase"

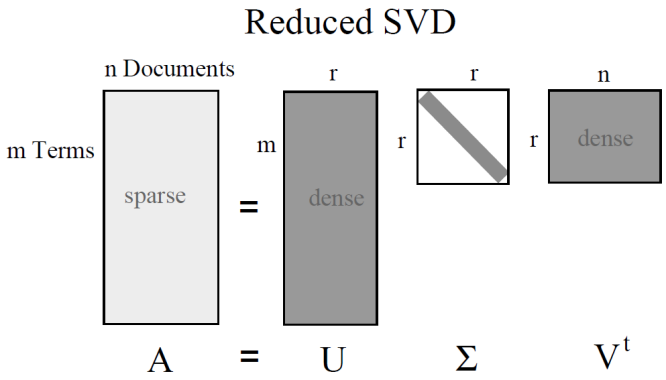


Figure 2.3: Term-Doc Matrix reduction with SVD

orthogonal matrices (with respective sizes of $terms \times r$ and $r \times docs$ where $r \leq docs$) whose columns form respectively the left singular vectors and right singular vectors. Σ is a diagonal matrix whose values are called singular values and are sorted in ascending order. Up to that point the idea is to select $k \leq r$ singular vectors with the highest singular values, so that the dimensionality is reduced while loosing as less information as possible.

SVD is very close to PCA analysis, the difference between both is that PCA is performed on the covariance matrix while SVD is performed on raw data. Due to this difference, SVD and PCA reach slightly different results (see figure 2.4[2]).

Topic Identification The main difference between topics and clusters is that clusters are generally exclusive, so a document belongs to one and only one cluster while a document can belong to multiple topics. SAS Institute (2014) defines a topic as "a subject or theme or idea that occurs in a document". The topic identification in SAS is black box, the only things we can know about it is that it uses the SVD and that it works with a kind of social network algorithm on the co-occurrence matrix (where words are "friends" when they often co-occur together). Since both the method proposed in this thesis and the SAS text topic node use SNA techniques, ones can expect to reach somewhat similar results. However, there is no clue about the use of the SVD in this process.

[2]Image taken from Albright (2004)

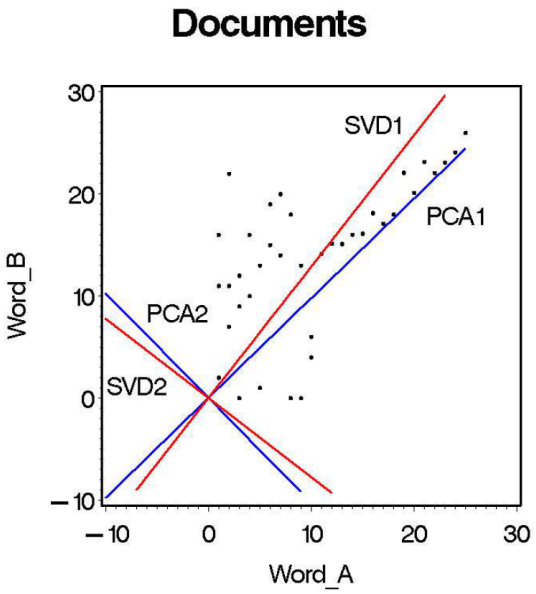


Figure 2.4: Term-Doc Matrix reduction comparison of SVD and PCA

Chapter 3

Social Network Analysis

This section defines what Social Network Analysis (SNA) is and describes some of the main techniques of the field. According to Feldman & Sanger (2007), a social network can be defined as "a set of entities and a set of relationships between them". Entities can be people, objects, organizations, events... while the set of the kinds of relationships is very broad (belonging, friendship, family links...). According to Otte & Rousseau (2002), SNA is "a broad strategy for investigating social structures". So SNA is a set of means to discover the structure and the characteristics of social networks. It includes especially concepts and some techniques from graph theory (among which centrality measurements, shortest path and connectivity). Furthermore, according to some authors (eg Duijn & Klerks (2014)), Social Network Analysis (SNA) can be defined as a domain crossing Graph Theory and Data Mining.

The biggest challenge of such algorithms is their algorithmic complexity, many graph search algorithms are NP-hard or even NP-Complete. It means they can not be solved in a polynomial time. So some authors such as Cuvelier & Aufaure (2011) and Skhiri & Jouili (2012) try to find solutions to handle this complexity. The latter uses the parallelism property of this kind of algorithms to run them on platforms like Hadoop.

3.1 Basic concepts from graph theory

This section describe briefly some basics from graph theory.

As defined in Lambiotte & Tabourier (2012) a graph is connected if a path

exists between each pair of nodes. A connected component is a maximal connected subgraph. So intuitively, it corresponds to bunch of connected nodes which has no links with exterior.

The same authors define an adjacency matrix as: For a graph G with n nodes $v_i, ..., v_n$, the adjacency matrix of G is a nxm matrix where ij element is the number of edges between nodes v_i and v_j.

Geodesic path between two nodes is defined by Bouttier *et al.* (2003) as the distance of the shortest path between two nodes. If nodes are in different connected components, then the geodesic distance is set to infinite.

3.2 Centrality Measurements

Feldman & Sanger (2007) and Lieu *et al.* (2014) focus on centrality measurements, with examples based on the networks of 09/11's hijackers (see figure 3.2). Centralities are measurements which aim to find central actors within a network.

Degree Centrality of a node is simply its number of connected edges. So it means in a social network, someone's number of friends.

Closeness Centrality of a node is its geodesic distance from all the other entities of the network. It corresponds, in a social network, to how a person knows everybody from the network through the least possible number of intermediates.

Betweeness Centrality represents the number of geodesic paths between pairs of nodes on which a node is situated. From an information flow point of view, it could be seen as the quantity of information which passes through an entity.

Eigenvector Centrality is the principal eigenvector of the adjancency matrix. It is a concept between degree and betweeness centralities. It represents how someone is on the information flow of well connected entities.

Power Centrality is a normalized version of eigenvector centrality, so the explanation is the same.

Network Centrality is another kind of measurement, since it measures the connectivity of the whole network. The measure ranges from zero to one, where zero represents a non-centralized network (graph looking like a circle) and one represents a centralized one (a graph in a star shape). A highly centralized network is somewhat built following a hub and spoke architecture.

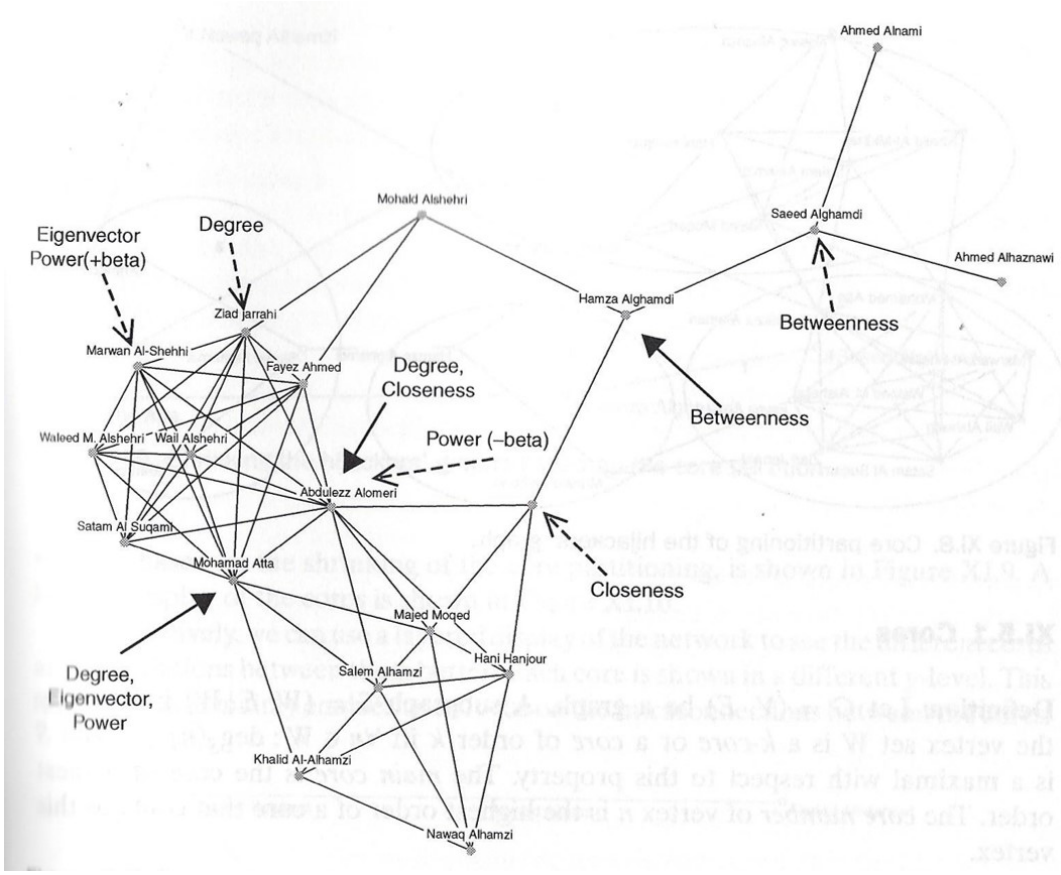


Figure 3.1: Network of the terrorists of 9/11th attacks with the different centrality measurements

3.3 Community detection

Community detection, which is also known as graph clustering aims to find subgraphs (groups of nodes inside graphs) which are highly connected inside themselves, but less connected between each others. A well-known example of this technique, presented in Cuvelier & Aufaure (2011), is the Zachary's Karate Club. It is a Karate club in USA, where friendships and connections between members were observed for three years, afterward a graph was built showing the different communities inside the club. There are many other applications of these methods outside the social network area like VLSI (Very Large Scale Integration) layout design, as explained in Ghosal *et al.* (2005), which aims to minimize the costs of chips. Since intra-board connections are less expensive than inter-board connections, components are clustered and each cluster is built on different boards.

There are several kinds of community detection algorithms. The article from Lancichinetti & Fortunato (2009) provides a benchmark of the most well-known ones. This chapter focuses mainly on two techniques: the fast unfolding method from Blondel *et al.* (2008) and the label propagation method presented in Raghavan *et al.* (2007).

The fast unfolding method, also known as Louvain's method, aims to optimize modularity. As presented in their paper (Blondel *et al.* (2008)), modularity is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} [A_{i,j} - \frac{k_i k_j}{2m}] \delta(c_i, c_j)$$

where A_{ij} is the weight of the edge between nodes i and j, k_i is the sum of the weights of the edges adjacent to i, m is the sum of the weight of all edges and $\delta(c_i, c_j)$ is a binary function returning 1 if both i and j belong to the same community, 0 otherwise. So broadly speaking, modularity is high when edges of high weight connect nodes within the same community.

1. Initialization: each node belongs to its own community;

2. Move each node to its neighboring community which increases modularity the most until modularity can't be improved further;

3. Build a network where communities found at the end of step two are nodes, and

repeat these steps until modularity can't be improved further or if a time/iteration threshold has been reached.

The label propagation method, while somewhat similar, does not aim to maximize the modularity. As explained in Raghavan *et al.* (2007), the algorithm works as follow:

1. Initialization: each node has its own label;
2. Each node updates its label with the one shared by a maximum of its neighbors.

In opposition to Louvain's method, label propagation method can cope with directed links (also known as "arcs").

For both method, it is possible to tune the algorithm so that it produces more or less communities. The parameter used for this is called "resolution limit". Broadly speaking the resolution of a community is the strength of its intra-community links to be able to merge them, as explained in Ronhovde & Nussinov (2010). The resolution limit requires communities to have a minimum resolution. In both Louvain and Label propagation, the resolution is directly linked to density, but their interpretation differs a bit. For Louvain, "Two communities are merged if the sum of the weights of intercommunity links is at least x times the expected value of the same sum if the graph was reconfigured randomly" (Inc. (2015)). For the Label Propagation algorithm, the resolution limit represents the minimal density of communities. Pictures 3.3[1] and 3.3[2] show how results differ given a different resolution limit.

In general, label propagation algorithm creates more of smaller communities than Louvain's algorithm. Test made by author on another project on the difference between both algorithm are presented in appendix A.

[1]Image taken from Inc. (2015)
[2]Image taken from Inc. (2015)

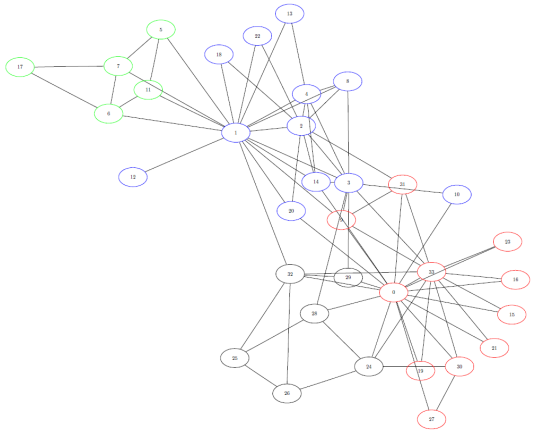


Figure 3.2: Community detection with Louvain's algorithm (resolution 1)

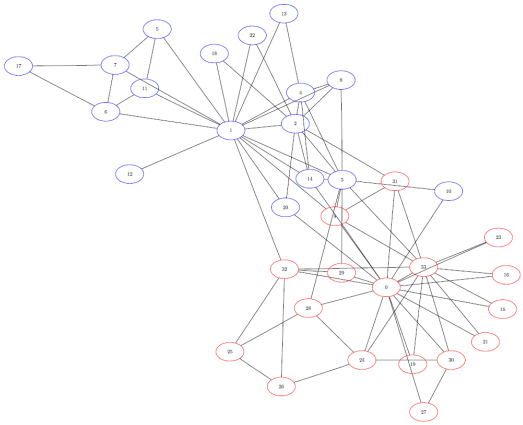


Figure 3.3: Community detection with Louvain's algorithm (resolution 0.5)

Part II

Experimentation

Chapter 4

Exploration with existing methods

This chapter gives an overview of results from existing Text Mining exploration methods. The movie dataset[1] includes 1527 movie reviews from a movie reviewer's website [2]. These reviews are written in English and somewhat express the feelings of the author concerning the movies. The dataset contains several target variables like the genre of the movie, so that it is possible to use this dataset for prediction, too. However, these variables were discarded to focus on the exploration. It was decided to focus on this dataset only, rather than using several ones. Indeed, since the focus of this thesis is to provide a new method in unsupervised learning, the assessment of the results is only qualitative. So a good assessment requires knowing the dataset well.

This dataset is interesting because it is a difficult one for such a text mining task. There is only one main topic: the movie topic, in opposition to other text collections or text datasets, such as the well-known WeatherAnimalsSports dataset[3] (which contains these three main topics) or even the dataset related to the Unabomber[4] (which contains textual documents from different authors). So the method has to be able to extract sub-topics in order to get findings of interest. Obtaining good

[1] Taken from the course SAS Institute (2014)
[2] http://www.susangranger.com/
[3] Taken from the course SAS Institute (2014)
[4] Taken from the course SAS Institute (2014)

results for this task means getting a segmentation of the films related to their content (science-fiction, action, war, romantic...). The method has to avoid creating segments with generic movie words.

The aim of this "regular" exploration is not to find the best parameters for clustering or topics analysis. It is only to find bag-of-words which are consistent regarding the objective presented above. Finding such results from the text clustering or the topic extraction took a lot of iterations. Beside an example showing how "good" clusters are selected from a text custering, only the consolidated list of clusters/topics with some of their corresponding keywords are presented (more results are presented in appendix B).

Nevertheless, if the objective is not to find the best parameters for clustering or text topics, good parameters for the NLP can be used as a starting point in the NLP part of the developed method. That is the reason why the two parameters of a minimum number of documents and the term weight parameters are explained more in details.

4.1 Natural Language Processing

This part of the exploration was needed in order to do further analysis on it, but overall it was a good exercise to find parameters to use as a starting point in the NLP part of the proposed method. The NLP process in SAS® is split in two different nodes: the text parsing one and the text filter one.

The first one carries out the POS tagging, the noun groups detection (n-grams detection), entity extraction (either standard entities or user defined entities), the POS rejection (eg auxiliaries, conjunctions, determinants, interjections, particles, prepositions, pronouns), attributes rejection (eg numbers and punctuation). This node also takes care of the stemming and the synonyms merging (synonyms come from the standard synonyms table SASHELP.ENGSYNMS), as well as the filtering of a stop/start list (in this exploration the standard stop list is used from SASHELP.ENGSTOP).

The second node, the text filter node, is probably the most important one of the rest of the exploration, as well as for the method developed in this thesis. The check spelling option, which in fact calls for a SAS procedure which checks for spelling, was not used for two main reasons: first, this process is resource demanding, then

Cluster ID	Descriptive Terms	Frequency	Percentage
1	gauge granger movie +writer +daughter +director +write +comedy +story +play +direct john +star back +young ...	197	13%
2	best +late +role +great +actor +base +begin +film first +character +play +look +director +year back ...	148	10%
3	+role +work +late +year +turn +plot +life +end +audience +man john +picture +begin +direct +character ...	762	50%
4	+movie +run +rate +show +kid +long +script +language violence +old +want +star +little +writer +good ...	420	28%

Figure 4.1: Example of text clustering (EM) on Movies dataset (using TF.IDF term weights)

assuming that the author writing reviews has a degree in journalism, it is reasonable to expect her texts to be misspelling-free. Then, it is the node where frequency weights (also known as local weights) as well as term weights (also known as global weights) are defined. Then it is also the node that filters the terms which are below the minimum number of documents threshold, as well as the maximum number of terms with the highest global weight to keep.

Several values for these parameters have been tested. The frequency weight was set to log since it is the standard value for this parameter and seeing that it does not influence the term weight computation. Furthermore, log is generally the best choice since it keeps the information from the frequency of the term (in opposition to binary) while countering effects from the skewed distribution of frequency (in opposition to identity). For term weight, mutual information has been discarded since the target variables of the dataset were not used. Both entropy and TF.IDF were tested with different cut off values for the minimum of documents a word has to belong to (2;4;10;15;30). According to the result quality with both text cluster and text topic nodes, the best value for this parameter seems to range between 10 and 15. Indeed, a small value for this parameter seems to increase the size of the term-doc matrix and decrease SVD performances, while a high value leads to reject discriminative words. Entropy seems to reach better results than TF.IDF with the same set of parameters, but also with different parameters. Figure 4.1 shows results from a EM clustering with TF.IDF term weights: the different clusters contain only generic words from the movie theme (”actor”, ”film”,”play”,”show”, etc.). It was tried with different clustering parameters (SVD parameters, clustering algorithm, number of clusters), all performed similarly: no interesting clusters, or a few regarding the whole analysis.

4.2 Text clustering and Text Topics

The text clustering node handles both the singular value decomposition and the clustering algorithm. The SVD requires two different parameters: the resolution and the maximum number of dimensions. The last one is the maximum size of the Σ matrix. The resolution parameter (high, medium, low) corresponds to the proportion of singular values to keep. For example, "low" as a resolution parameter has a value of 66.67%, which means that the first singular values whose sum is greater than or equal to 66.67% of the sum of all singular values are kept, while a "high" resolution takes all singular values. The default parameters (maximum 100 dimensions and low resolution) reach the best results. Indeed, the SVD on the movie dataset with these parameters creates 48 new dimensions, so the SVD does not use the maximal number of dimensions allowed.

It seems that changing the resolution value "medium" leads to reach good results with expectation-maximization algorithm (see figure 4.2). Indeed, if each topic does not represent a movie theme, some seem to do. These are the number 3, 8, 11, 13, 14, 15, 16 and 19 (see figure 4.2). The rest of the clusters do not have consistent bag-of-words describing them. Furthermore, if the topics cited above seem to represent movie themes or categories, there are some words for which the link with the theme is not clear and some that do not have any link with the precise theme but rather with the general movie theme. For example, for cluster 3 and 8, there are some noise words ("last", "find", "first", "minute", "acceptable", "release", "viewer", "minute", "sequence" and "cinema") which do not give information about the theme or category but which come from the general movie theme. There are also other words which are somewhat meaningful but for which the link with the topic is not clear: "soldier", "effect" (which may come from "special effect"), "action", "motion" (which may come from "motion picture(s)"). For the rest of the clusters, only the words directly related to the category they belong to are presented.

- Clusters 3 and 8 : these clusters seem to be described by horror and science-fiction keywords ("alien", "earth", "human", "horror", "frighten", "scary" and "killer").

- Cluster 11 : this cluster probably represents comedy movies. Indeed, although there are only a few interesting words, the first ones seem to refer to the comedy

Cluster ID	Descriptive Terms	Frequency	Percentage
1	+motion +sequence +viewer +effect +moment +picture +place +fall +lead +minute +battle +performance +begin ...	84	6%
2	+run +rate +language +teenager acceptable +kid sexual +old +recommend violence +long cinematography fine ...	194	13%
3	+alien earth +human +effect horror +frighten last +find +horror +soldier first +minute acceptable +scary +action ...	40	3%
4	+win down +murder academy +help james +sequence +turn allen +nominate +performance +force supporting fi...	46	3%
5	+order hollywood john +force +viewer +hand +protagonist +lead +reason +killer +war bond war +kill +great ...	23	2%
6	+chase +action +sequence +plot +agent bond +release +villain +viewer +minute far +picture +audience +well +...	82	5%
7	hollywood +protagonist +order +late +plot +reason +hand +direct +viewer +great +villain +actor +base +role exc...	96	6%
8	horror +horror +scary +frighten +release +motion +human +viewer +minute +effect +sequence +killer first cinem...	37	2%
9	movie gauge granger +daughter +writer +director romantic +write +story +comedy +play +direct woody john +yo...	175	11%
10	allen woody jeffrey +age bond +cast +funny +comedy first pg serial +little down +old +time ...	32	2%
11	+joke +laugh +comedy +funny +moment far +minute +motion supporting +thing +work +viewer +cast +plot +rele...	92	6%
12	+young +woman +man +guy +friend +people +place +audience +life back +daughter +bond far +kid +look ...	138	9%
13	military +effect war +action +war +villain +alien +order +world earth hollywood +human +protagonist connery all...	47	3%
14	+war war +battle military +soldier +army +force +order +oscar +human +base +late hollywood +kill +win ...	52	3%
15	romantic +comedy +fall +love supporting +woman +motion +moment +day +friend +role +funny +end +laugh +th...	67	4%
16	best academy +nominate awards actor actress picture +win +oscar supporting cinematography +award excellen...	31	2%
17	+kid +rate +age +son jeffrey acceptable pg +run +funny +recommend +little +language +human +movie +laugh ...	101	7%
18	+show +rate +recommend acting +award +teenager fine +script violence +start cinematography +bad +people +...	134	9%
19	+killer serial +murder +agent +kill +help +villain +base sexual violence +teenager +frighten connery +release ac...	35	2%
20	+bond bond connery james pg +agent jeffrey +chase +villain fine sexual beautiful +son +sequence violence ...	21	1%

Figure 4.2: Text clustering (medium SVD resolution) with Movies dataset

category (”joke”, ”laugh”, ”comedy” and ”funny”).

- Clusters 13 and 14 : these two clusters describe the war movie category with the following keywords: ”military”, ”war”, ”protagonist”, ”battle”, ”soldier”, ”army”, ”force” and ”kill”.

- Cluster 15 : the two first words describing the cluster probably describe the romantic comedy category. Indeed, besides these two words there are ”love”, ”woman”, ”friend”, ”funny” and ”laugh”.

- Cluster 16 : this cluster does not represent a movie theme, but rather the award theme (”best”, ”academy”,”nominate”, ”awards”, ”win”, ”oscar” and ”award”).

- Cluster 19 : this cluster probably matches with the thriller category with words such as ”killer”, ”serial”, ”murder”, ”agent”, ”kill”, ”villain”, ”sexual violence” and ”frighten”.

The parameters of the clustering algorithm are the algorithm itself (Expectation-Maximization or hierarchical) and the number of clusters (that can be maximal or exact). Both Expectation-Maximization (EM) and hierarchical clustering were tested. However, test with the hierarchical clustering produced poorer results than

Cluster ID	Descriptive Terms	Frequency	Percentage
18	horror +person young +death +viewer +horror minutes supernatural cinematography +child simply +mother works +credit +mind +matter +nomination particularly +right +fact +motion +help finds +strong +relationship ...	48	3%
22	+motion +viewer performances +care moments +act +end +moment lives +screenplay +person +home +child things +relationship simply written +woman +picture +reason characters +work ends wants +mind +life +t...	74	5%
23	sexual +sex rated +guy runs +comedy acceptable +line +love wants best women +strong +language american works fun last violence +audience kids teenagers +long +look john +play +drug +funny +bad +man	36	2%
26	eddie murphy +comedy hilarious amusing +funny movie gauge granger jokes supporting york +work +sweet plays tv adam +love robert +help humor finds +little dr laughs +family +know things sexual john	56	4%
27	exciting sequences visual nearly effects +viewer +premise +action +sequence +opening +motion +dialogue far +genre +picture +work +care +help +fight simply +plot directed +reason +character +well three worth +lo...	57	4%
29	fiction +author hollywood +cinema +result played +book original films directed interesting +plot +novel +hand +order begins +reason looks +great effects +version today based later actors especially +film +world +go...	70	5%
31	+show +recommend cinematography nudity 1/2 +award full wonderful acting rated +sex fine sets +problem runs excellent lines +great +script teenagers violence +long found starts +well especially +want +people sc...	115	8%
32	ryan +love romantic recent starts +half +well +comedy +picture +funny less +man +woman serious moments next right rated +recommend wants first +performance +cute runs fun +set +end playing +script +day	29	2%
35	bad things john +kill +car +bad +reason minutes back scenes down +thing far +people +plot +point fun +big +screen plays directed films michael +know years romantic +daughter +action based finally	40	3%
36	academy awards nominated actor actress picture won best cinematography supporting +oscar leads excellent +production +film later +cast +actor +well men +character films robert +place +young violence times +sc...	49	3%
37	bond +bond connery +innuendo gadgets jeffrey james pg +agent +chase +villain beautiful sexual fine kids british rated +car runs dr +son violence films women series +time +action +world +movie +set	35	2%
38	gauge granger movie +mother emotional +daughter peter tv +writer +story meets year-old +comedy +father +director begins parents romantic michael +drama +school plays +woman +girl particularly +young directed ...	113	7%
40	runs kids rated +son jeffrey pg +dog ages 1/2 +age acceptable +recommend +sweet +show +cute +family parents +little +funny humor +big violence +script +movie especially +bad best +good music +father	67	4%
41	rated runs +president +language fine +right entire +head starts general +oscar plays military finally kids +show +heart +picture +well +home best +screen +performance back +want better first +script violence +war	26	2%
44	+husband particularly +woman +novel +relationship +wife movie gauge granger +crime +girl +story +life +drama +father +man looks york +friend +young +performance +job women three +family +audience +director ...	100	7%
48	acceptable english subtitles 'english subtitles' rated runs pg-13 jeffrey kids fighting +language violence +age +long +version teenagers watching +fight +son +old +large +action +picture worth sequences stars +movie...	41	3%
49	runs rated +language teenagers violence acceptable older +drug nudity +sex starts +school +bad +car kids +script +high +black sexual +strong +movie +long +scene +show +writer plays +know +want +figure pg-13	111	7%
52	runs rated pg-13 1/2 nudity +language acceptable +recommend +show kids +funny violence jeffrey +well +long +sex +movie +star +son +comedy humor bad +school +script sexual +year +big +good +age first	90	6%
53	allen +thriller +motion hollywood beautiful +plot +relationship films +role years written simply begins +help +love +thing moments later horror characters +screenplay +strong directed +character +play +action serial +...	69	5%
57	+blood full violence +action fine +death teenagers +recommend effects +well +good fun movies nudity +people rated +reason +know +script minutes cinematography runs +performance +look +plot acting scenes +fe...	44	3%
59	effects war led actors +order +war +hand +history films +reason hollywood later military +small american +plot different +action +great last movies +set today general +bad +genre +look +group played +end	113	7%
62	+nomination +admire news +screening +chief +press +image +president +book +song fine +government rated sets pg +language 'a bit' attempts +figure runs able +writer allen +day ends pg-13 +talk +script +order ...	7	0%
66	+'serial killer' serial +killer lambs +thriller silence supernatural murders horror +horror +evil +child +direction +credit +screen +murder gauge granger +book movie +matter +boy +star +mind +help based turns +writer ...	16	1%
68	+school +relationship +woman simply +short friends +mother +life +friend moments far lives +love +home sexual +guy women down romantic money +know +high +day +picture +cast +find +place +work +job +drama	75	5%
71	laughs adam +laugh +sweet jokes +comedy +premise fun +funny +entertainment entertaining +material far +feature minutes +screenplay hilarious +genre +work humor +star directed major tv written +big +group sup...	46	3%

Figure 4.3: Text clustering with hierarchical clustering algorithm on Movies dataset

those from EM clustering (see figure 4.2). The EM clustering implemented is the one described in Moon (1996).

The parameter defining if the number of clusters chosen is the maximum number or the exact number, is set by default to maximum. It appears that changing it into "exact" leads to get "good" results even with low SVD resolution (see figure B.1). Text topics with standard features also directly reach interesting results (see figure B.2) and playing with the number of topics parameter leads to find other interesting categories. The selection of "good" clusters among others is not presented more in details in this section, but interesting findings that led to the consolidated list presented in the next section are presented in appendix B.

4.3 Findings of the exploration

An exploration of the dataset gives interesting findings. First, some parameters' values which seem optimal to start from: the minimum number of documents a term has to belong to (between ten and fifteen) and the best term weight method (entropy). The findings of the different topics analyses and text clusterings (see appendix B) are consolidated in the following categories (containing the corresponding words):

- Comedy movies : "comedy", "joke", "funny", "humor", "laugh" and "comedic"
- Romantic movies : "relationship", "romance", "romantic", "care", "friend", "actress", "love", "woman", "date", "man", "money", "girl", "college", "young"

, "guy", "meet", "parent", "people", "wife", "mother", "home", "job" and "school";

- Romantic comedy : "love", "woman", "friend", "funny" and "laugh".
- Action movies : "chase", "stunt", "action", "car" and "fight", "gangster", "crime", "death", "fbi", "academy", "bond", "connery" and "james", "gadget", "agent", "chase", "villain", "sexual", "british", "car", "violence", "kill", "british", "nudity", "special effect".
- War movies : "war", "battle", "soldier", "protagonist", "army", "war", "state", "military", "american", "history", "fight", "government", "german", "force", "mission", "die" and "kill". It could be possible to have more specific war movies : "jew", "jewish", "holocaust", "soldier" and "nazi", "camp", "german", "death", "history
- Thriller movies : "protagonist", "thriller", "detective", "villain", "crime", "murder", "serial killer", "serial", "killer", "fbi", "agent", "law", "murder", "detective", "kill", "cop", "prison", "sexual violence", "frighten", "car", "crash", "heist", "gang" and "criminal".
- Sci-Fi movies : "science", "fiction", "special effect", "effect", "human", "space", "future", "special", "earth", "Spielberg", "astronaut", "space", "program", "earth", "mission", "crew", "government", "future", "gadget", "alien", ", ship" and "machine".
- Musical movies : "song", "musical", "sing", "music" and "dance".
- Horror movies : "horror", "frighten" and "scary".
- Sports movies : "football", "team", "coach", "player" and "game".
- Political movies : "president", "political", "government" and "conspiracy".
- Teenager movies : "school", "student", "girl", "college" and "parent".
- Disney movies : "Disney", "voice", "animal" and "kid"
- Family movies : "family", "daughter", "mother", "child" and "parent"

- Law movies : "lawyer", "murder", "case" and "attorney".
- History movie : "jew", "camp", "german", "prisoner" and "jewish".
- "Martial arts" movies : "stunt", "chan", "Jackie", "martial", "kong" and "hong".
- Adventures : "Indiana", "Jones" and "adventures".
- Award nomination : "nomination", "award", "academy", "nominate", "best actor", "supporting", "actor", "oscar". This topic is not a movie category, but it is a sub topic in the movie theme.
- Pegi rates: "rate", "language", "acceptable", "sexual", "teenager", "pg-13", "nudity". This topic is not a movie category, but it is a sub topic in the movie theme.

The aim of these consolidated categories is to find out if the developed method is able to create automatically communities with their words or not.

Chapter 5

Data Preparation

This chapter focuses on the preparation of the data before building the network. This chapter does not take into account the very first step: data import and conversion, since the datasets used was already SAS dataset format. Some preprocessing techniques from Natural Language Processing are used in order to parse the documents, to extract concepts and to filter them. Then, some extra filters are used in order to avoid some obvious relations between concepts. Finally, the data is transformed in transactional data where transactions represent presence of a word in a document, paragraph or sentence.

5.1 Natural Language Processing with HPTMINE

Natural Language Processing (NLP) is still very important in the developed method because the textual data has to be parsed (see section 2.2.1) to extract concepts and useless words have to be removed (stopwords, some grammatical categories, some kinds of entities...). For the NLP, the procedure HPTMINE has been chosen. This procedure is the high-performance SAS procedure that handles text mining problems (Inc. (2016)). In comparison to the regular procedures, this one is multi-threaded, multi-processor, distributed (when used in a grid environment) and in-memory. Furthermore, it covers all the NLP needs: term-doc matrix transformation, stemming, POS tagging, synonyms, terms weighting, singular value decomposition... Then, Ames (2016) shows how easy it is to extract concepts (especially the entities in the article) and re-use the output datasets afterward.

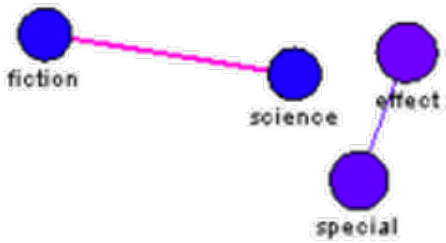


Figure 5.1: Multi-words links

The liste 5.1 shows HPTMINE structure. It automatically uses the input dataset which is plugged into the node (via the macro &EM_import_data which contains the dataset name imported by the flow), as well as the variables which have roles of ID (%EM_ID) and Text (%EM_TEXT). Then, the default stopwords list defined for English is used (stop S̄ASHELP.ENGSTOP where SASHELP is the library in which the ENGSTOP data set is located), as well as the default synonyms list (syn=SASHELP.ENGSYNMS). As explained in section 2.2.3, stopwords are words useless to catch the meaning of a phrase and they increase noise. Then, as explained in section 2.2.4, consolidating the synonyms allows to reduce the information spread into several words. These are two standard techniques in Text Mining. The following Part-Of-Speech (POS) have been discarded: auxiliaries, conjunctions, determiners, interjections, particles, prepositions, pronouns and numerical (with the statement: "select ignore"). These POS do not add information about documents themes and it is also standard to reject these POS in Text Mining. Indeed, "he", "the", "or"... do not provide information on the topic.

Then the standard entities are extracted (eg Names, Noun groups, miscellaneous propositions, locations...) for two main reasons: first to filter some entities later and then to retrieve multi-words concepts (noun groups, names...). This is done with the statement "parse entities s̄td". Indeed, some useless entities, such as time, duration or currency, were initially adding noise to the network by adding meaningless links between other nodes. Then, the multi-words terms became an issue encountered in the first attempts to build the network: the terms were split in two different linked nodes rather than having one multi-word node (see figure 5.1 where science is linked with fiction and special with effect).

The local weight (aka term frequency) has been set at log ("cellwgt = log"), but this parameter is not really important since it was not used to filter the words. The global weight (term weight) has been set to entropy ("termwgt = &glob" where &glob is a macro referring to entropy). Since it is not a supervised learning task, the mutual information criterion had to be discarded. Then, regarding TF.IDF, entropy has several advantages over it. First, in opposition to TF.IDF, entropy is bounded ([0,1]), while TF.IDF is not ([1,∞[) so it is more difficult to set a filter. Moreover, it appeared in the exploratory analysis that entropy preformed better. A filter is applied on the term weight, rejecting concepts with a term weight below 0.20. This threshold seems to be optimal and was found through experimentation. Values from 0 to 0.30 were tested, and it appeared that words below 0.20 add mostly noise (as seen during experimentation). Furthermore, an analysis of weights of interesting rules showed clearly that only words with weight above 0.20 were interesting. This analysis was carried out by manually tagged interesting rules as meaningful (see appendix 6.3).

Listing 5.1: HPTMINE structure

```
proc hptmine data=&EM_IMPORT_DATA;
        doc_id  %EM_ID;
        variables       %EM_TEXT;
        parse entities  = std
        stop    = SASHELP.ENGSTOP
        syn     = SASHELP.ENGSYNMS
        outterms        = terms
        outpos          = pos
        reducef         = &minDocs
        cellwgt         = log
        termwgt         = &glob;
        select "Aux" "Conj" "Det" "Interj"
                "Part" "Prep" "Pron" "Num" / ignore;
run;
```

5.2 Filters and consolidation

This section describes the different filters and consolidation methods applied to the dataset from NLP. These filters aim first to remove noise for the rest of the analysis, which leaded to poorer results in the following steps, but also to improve performances of the next steps, and especially for the association analysis which is highly resource demanding.

The filter on the minimum number of documents a word has to lay in, which is a HPTMINE parameter ("reducef = &minDocs" where &minDocs is a macro defined outside the procedure), was set to 5. The default number is 4, which is quite low regarding the number of documents in the dataset (1527). Experimentation was run with higher values, with the aim to filter more words. However, to avoid rejecting words that, once consolidated, could add important information to the analysis, the threshold was set at 5. The reason why it had to be lowered is described later in this section. Moreover, the value of 5 has been validated through experimentation. On the other side, a maximum number of documents filter was set after results showed that some meaningless words (eg run, rate...) had a high degree in the network. This parameter was first set to 500, but further experimentation showed that 200 is a better value for this dataset (see appendix 6.3).

A consolidation of stemmed words was applied. If the main objective of stemming is to avoid rejecting words, by proposing their radical, the aim is not to get rid of word derivatives. Indeed, if the derivative of a given word has a weight sufficiently high, as well as the radical, both can be kept since a derivative can add some information that can be handled in SVD. However, in this analysis, all words have been consolidated to their radical for several reasons. First, the information is spread in several concepts, so several rules with a lower lift or confidence, so it is likelier to be discarded. While these concepts could produce better rules once consolidated. This assumption has been validated through experiments. Then it avoids creating obvious relations between words radical and their derivatives. Indeed, such obvious links take place in the final graph which is limited to a thousand nodes for user readiness [1]..

Then, another issue related to consolidation arose through experimentation. In the created graphs, some words were linked with another node of the same word (eg

[1] This limit is really the extreme limit. The graph gets harder to read beyond five hundreds nodes.

"run" with "run", "fight" with "fight"). This is due to the fact that some concepts can belong to different POS groups (eg "to fight" is the verb and "a fight" is a noun). While this high correlation is easily handled by SVD (such kinds of highly correlated concepts will be caught by the same singular vector), this leads to losses of information in the developed method. Moreover, as explained upper, it also increases the number of nodes in the graph. Furthermore, since the rule generation procedure keeps only the concepts (and not also the term ID), it created further issues when joining the rules with terms table (because the concept is not unique so the join created the Cartesian product of matching terms). So matching terms were merged into a consolidated one, whose weight was the weighted average of original words weights. This choice does not affect the filtered words since a term weight filter is applied earlier. The weight is only used later to analyze the optimal weight threshold (see appendix 6.3). This consolidation is the reason why the minimum number of documents threshold has been set to five, while the objective was to select a greater one. For example, if the verb "fight" and the noun "fight" appear both in five different documents, then the consolidate concept "fight" appears in ten documents.

Finally, it appeared that multi-word concepts were linked to words composing themselves, creating cliques in the graph (see figure 5.2). These cliques leaded to the same issues explained upper: information spread (so loss) and room for meaningful concepts is lost. So it was decided to remove the words from transactions where a multi-word to which they belong appears. It can be argued that removing such words leads to lose information, which is true, that is why another alternative was explored: removing only the occurrences of single words that were redundant with their occurrences in the multi-word term. For example, if "science-fiction" appears four times in a document, and "science" six times, then, rather than removing science, "science" would stay in the document but with the corresponding count variable decremented to one. However, if this alternative seems obvious, it is not because of multi-words terms with more than two single word terms. Indeed, some sub-words from a long multi-words term can overlap, which leads to negative values in some words counts (eg in "academy award nomination" there are "academy award" and "award nomination" concepts, decrement sub-concepts would decrement "award" twice).

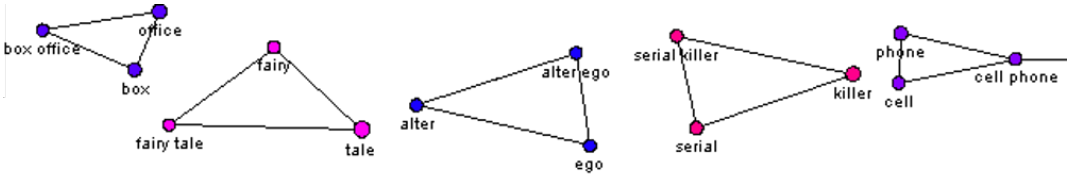


Figure 5.2: Examples of bi-gram cliques

5.3 "Transactionalization" of the data

This section briefly explains how the data from previous analysis is turned into transactional data. This step is very important because it transforms the data into the format required for association analysis. In this format, each line will associate a word with its frequency to a transaction (eg document).

The HPTMINE procedure can produce a lot of different datasets, among which two are useful: the first one with terms and the respecting term weight, occurrences, number of documents, etc and the other one details the position of each occurrence of each concept in every document. The idea is to use documents as transactions (see the example on figure 5.3). It is also possible to use either paragraph or sentence in place of document as transaction unit. Nevertheless, the reviews are quite short, so there are not a lot of paragraphs (if any) and using the sentence would lead to a too fine granularity (closer to syntactic links than semantic/topic related ones). It is interesting to note that using sentences as transaction unit could probably be useful for another task: finding n-grams. Indeed, first results showed a lot of multiwords relations (such in figure 5.1) and these results could probably be improved by refining the analysis by sentence, since parts of multi-words are necessarily in the same sentence as multi-words themselves. However, it was not the objective of this thesis, so sentence was used as transaction unit. A count variable was also created, although it was not used in the current analysis.

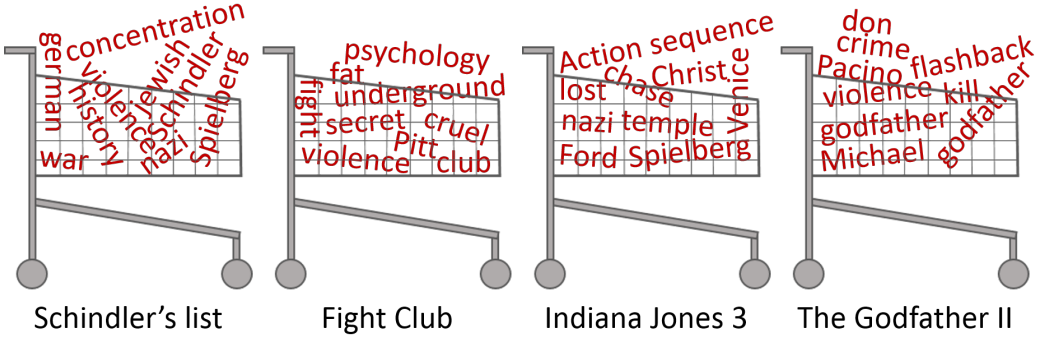


Figure 5.3: Data transactionalization

Chapter 6

Network creation

This chapter describes how the network is built from transactional data. The first idea was to multiply the term-docs matrix with its transposed version, so that it would result in a kind of co-occurrence matrix. However this method would have resulted in a very dense graph with many cliques (each document would create a clique with the words present in it), which is difficult to handle with community detection algorithms. So an alternative was proposed by this thesis promoters: using frequent item set mining to get association rules between words, so that these rules could be turned into graph links.

6.1 Association Analysis

For the association analysis, the two procedures used in the "association node" were implemented: proc ASSOC and proc RULEGEN (whose structures are given in listing 6.1). The first one aims to discover set of related items (so items that often occur together in transactions), while the second one creates the rules lying in these sets of items. So proc ASSOC is the procedure doing market basket analysis (MBA), which is also called "frequent item set mining" (Hossain *et al.* (2014)).

Proc ASSOC requires a minimum support parameter ("support&support" where &support is a macro defined outside), expressed as an absolute number (so the number of transactions in which the rule occurs). This was set to 10, to be consistent with the minimum number of documents a word has to be present in (see section 5.2). It means that a rule should occur in at least 0.65% of all transactions, which is

quite low given that the objective is to find topics of the whole dataset. However, it is more cautious to take a low value and filter them later given that experimentation showed that some interesting rules had low support values.

Proc ASSOC also requires a maximum number of items to consider in rule generation ("items=2"). Since the aim is to turn the rules into links, it is not possible to choose another value than two. However, from a Text Mining perspective, rules with more than two items could provide an interesting insight on the data. For example, a three items rule could find that "*academy*"&"*award*" ⇒ "*nomination*".

Finally, proc RULEGEN requires a minimum confidence parameter ("minconf=&conf" in which &conf is a macro defined outside), for which optimal value was set to 10 (which is the default value). Experimentation showed that the default value performs better than higher or lower values.

The parameters "cust" and "target" are also very important. "cust DOCUMENT" is the parameter precising that the transaction variable is DOCUMENT, while "target TERM" specify that an item is a term.

Listing 6.1: Assoc and Rulegen structure

```
proc assoc data=&EM_IMPORT_TRANSACTION
        dmdbcat=catassoc out=datassoc
        items=2 support=&support;
cust DOCUMENT;
target TERM;
run;

proc rulegen out=&datrule
        minconf=&conf;
run;
```

An interesting plot created by the association analysis node based on proc RULEGEN generated data is the rule matrix. The rule matrix plots the confidence of each rule in a matrix whose rows are the left hand side of the rule and columns are the right hand side. So from a graph perspective it is a representation of the adjacency matrix where each position value is the confidence of the rule implying this link (which is represented by the color from blue to red, where blue is a low confidence and red a high one). It appeared through tests with different parameters that the

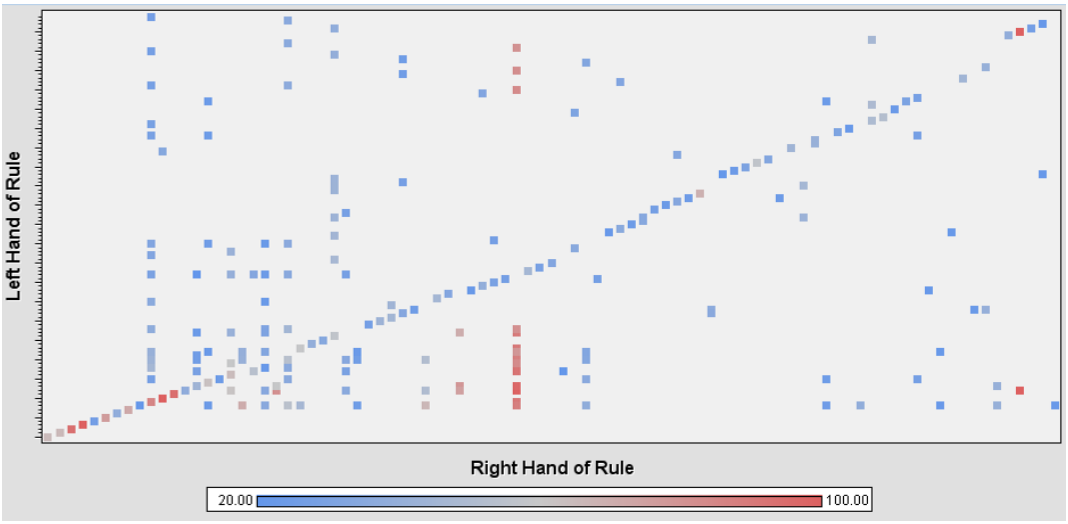


Figure 6.1: Data transactionalization

sparser this matrix is, the better the graph is (besides the diagonal elements). By taking, as an example, figure 6.1, ones can see that there are some vertical patterns. It means that a lot of different words imply a same word, so the node generated from that rule will have a high centrality degree. Indeed, by having a look to the graph generated with these rules (see figure 6.1), ones can see that some words have a high centrality degree. In particular, the red vertical line (or pattern) represents all the words that imply the word "rate". It is to tackle the issue underlying these vertical lines that a maximum number of documents threshold was introduced in section 5.2. Since red represents high confidence, it shows that is not a good feature to gauge a rule's pertinence. Indeed, confidence increases as support increases. However, the diagonal line is not a problem: it is related to how the matrix is built (it represent rules added while the elements of these rules were not yet present in the matrix). So the sparser that matrix is, the better the graph is. This assumption was validated since the rule matrix on figure 6.1 was built with the rules that generated the best topic communities.

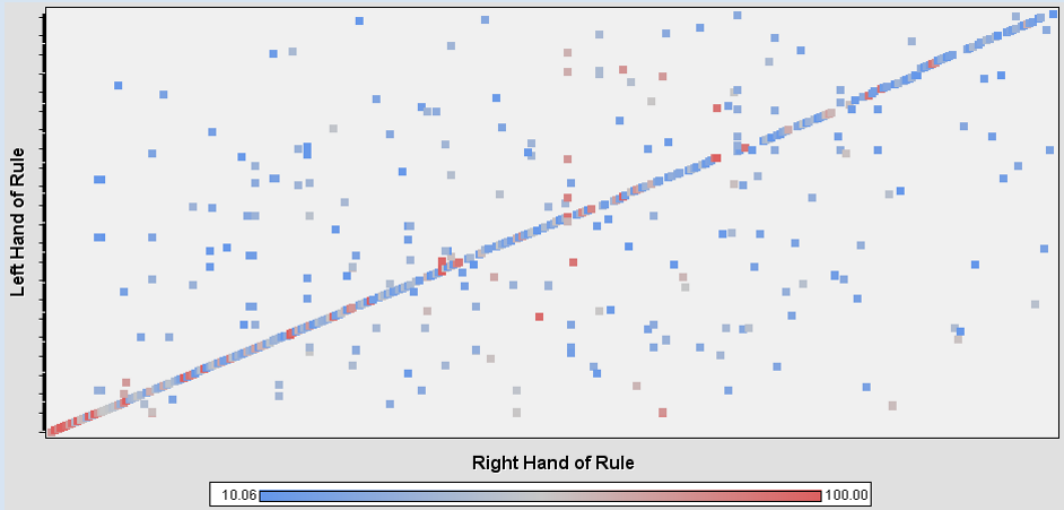


Figure 6.2: Data transactionalization

6.2 Filters and nodes and links creation

As explained in the previous section, it appeared that confidence does not carry pertinent information to select best rules. Since expected confidence, which is another measure created by proc RULEGEN, is the probability to find both words in the document collection, it is not of interest because the more both words occur, the higher the expected confidence. So, frequent words have more importance than the others, while it does not mean that they are discriminative (it is even the opposite relation: the more frequent, the less informative it is).

The default selection method of the most important rules in the association node is the descending lift. The user has to select a number of the best rules to display (200 by default), the ones with the highest lift being selected. Other features were tried (to make sure that using the highest lift rules generated the best graph) without success. It means that other features did not generate rules with meaningful words (as defined in section 4.3), while lift does not only generate rules with meaningful words, but also communities based on these rules which correspond to the topics described upper as, as an example, figure 6.2 shows for the comedy category. Moreover, 6.2, which plots proportion of flagged rules as meaningful against descending lift deciles (from

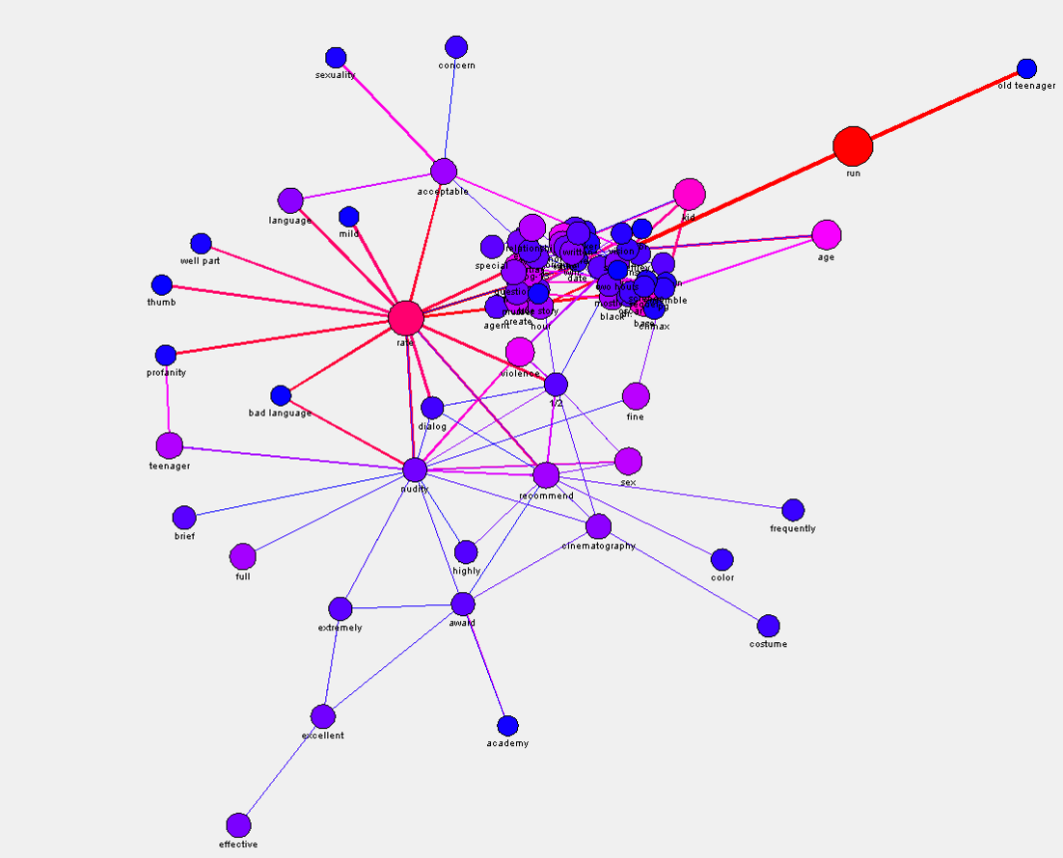


Figure 6.3: Data transactionalization

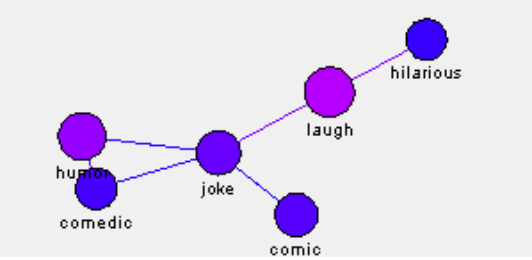


Figure 6.4: Comedy community with developed method

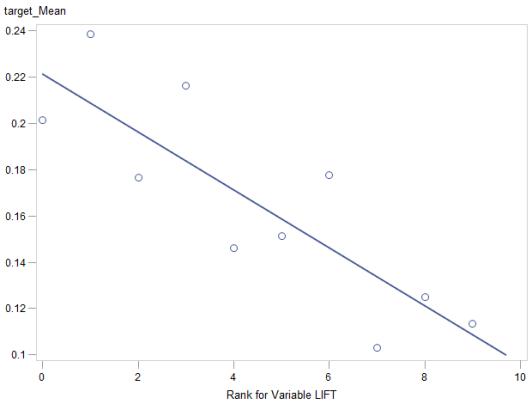


Figure 6.5: Proportion of flagged data by decreasing lift deciles

the analysis described in appendix 6.3), shows that the proportion of rules flagged as meaningful decreases as their lift decreases.

Furthermore, lift has an interesting property demonstrated in section 1.2.1: the lift is associative. It means that a rule in one direction has the same lift if the direction of the implication is reversed. Although the weight is not used in the network analysis part of the developed method, it could be easily used in further analysis since it would be easier to turn the rules, so from a directed graph to an undirected one (since both directions have the same weight).

Another idea to filter the rules was to merge both directions of rules so that it would have all the information about the relation of these elements. However, it appeared that only a few rules had a counterpart in the other direction due to some previous filters, among which no one was really interesting. The filters were soften

to increase the number of bi-directional rules, but it confirmed that bi-directionality is not necessarily interesting regarding the analysis.

6.3 Filters optimization

As explained in section 5.1, an analysis of rules attributes has been carried out. The objective of the analysis was to find the optimal thresholds for the different filters. So it was decided to add as much information as possible to the links and to manually flag meaningful links. The whole analysis was run with high acceptance thresholds in the different filters to avoid introducing a bias by bounding the analysis (through filters). Then the rules were sorted by decreasing lift, and the first 1500 rules were manually tagged as "meaningful" or not.

RULGEN generated the information described in section 1.2.1 (COUNT, SUPPORT, LIFT, CONF and EXP_CONF) and the rules where merged with terms dataset output from the NLP to add the following information:

- weight_from & weight_to : weight of words
- numdocs_from & numdocs_to : number of documents words belong to
- freq_from & freq_to : frequency of words

From which some features were computed:

- The minimum of weight_from & weight_to
- The minimum of numdocs_from & numdocs_to
- The minimum of freq_from & freq_to
- The maximum of numdocs_from & numdocs_to

However, they did not provide interesting insight about the optimal threshold, except minimum weight which confirmed that 0.20 was a good conservative threshold (see figure 6.3). Furthermore, if the basic objective was to create a model predicting, based on these features, if a rule was good or not, it appeared that such a model was not accurate enough. So rather than building a model, simple statistics (see array

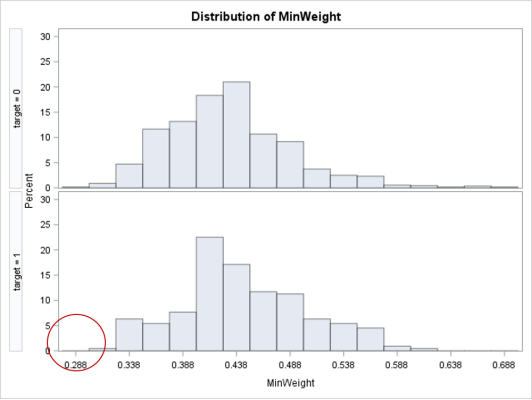


Figure 6.6: Minimum weight distribution of meaningful rules (target=1)

C.1 in appendix C) were used to select the optimal threshold to remove as much noise as possible without loosing any important information.

The rest of this analysis showed very interesting properties of "good rules". First, it seems to have a higher confidence than "bad rules" (see figure C), which was expected since it was showed in previous section than LIFT is linked with flagged rules and that CONF is a component of LIFT. Then, in one hand, it seems that the frequency of the right hand side of the rule should be higher than 25 (see figure C), so a word should imply a frequent word (not in terms of documents in which it is). On the other hand, it seems that good rules have a frequency of the left hand side's word of at most 216 (see figure C), so a word should not be too frequent to imply another. Then it seems that a word implying another should not be in more than 104 docs (see figure C), so "common" words which imply others do not give a good information (this could be expected, since random co-occurrences increase as frequency increases). Finally, a very interesting and unexpected insight is that, in opposition to unflagged rules, flagged ones have a higher weight in the left hand side than in the right hand side (see figures C and C). It means that a good rule is generally represented by a informative word implying a less informative one. t-test proved that this relation is significant at a p-value threshold of 0.001 (figure C shows the distribution of the difference between both). Furthermore, another t-test showed that weight_from mean is significantly higher in flagged rules (with a rejection threshold of 0.05) than in unflagged ones and weight_to mean is significantly lower

in flagged rules (with a rejection threshold of 0.05) than in unflagged ones.

Chapter 7

Network Analysis

This chapter presents results from the network analyses which are commented and compared to results from the exploration analysis (see section 4). The results of the analysis of the network through visualization are presented first. Then, the communities resulting from community detection algorithm are presented.

7.1 Network Visualization

In this section, results from the visualization are discussed. Before discussing the content of the results, it is interesting to first discuss the "container". The container in visualization is the algorithm used to draw the graph, as well as the number of links to keep in the graph, so that it is still readable. Then, it is important to discuss the impact of this parameter on the topics.

First, when comparing both graphs built with multi-level force algorithm (see figure 7.1) and spring force algorithm (see figure 7.1), ones could argue that spring force seems clearer because it is sparser. Nevertheless, in this case the multi-level force visualization is better for three reasons. First, connected components of more than two nodes are well separated in the graph. In fact, the fewer nodes there are, the less community detection algorithms are useful because the communities are in different connected components. So if they are well separated it is very easy to identify the different communities and, by extend, the different topics. Then, the biggest connected component is centered, and connected components of more than two nodes are "ordered" by size (the closer a connected component is from the

biggest component, the more nodes it consists of). Finally, such a graph is composed of many two nodes connected components (composed mainly of bigrams). Two words component add only noise to the analysis since it is almost impossible to guess a topic from a pair of nodes. In multi-level force algorithm, such bigrams are rendered in a kind of "needles box" (see figures 7.1 and 7.1), next to the biggest connected component, while in a spring force algorithms, all these bigrams are rendered quite randomly on the graph crossing edges from other connected components which makes it harder to analyze.

Then, the maximal number of links in a graph so that it is understandable seems to be one thousand. This limit is a bit subjective, but by having a look at figure 7.1, ones can see it is already quite dense, so increasing it would make it less understandable and the benefits of the visual interpretation of the method would be lost. Furthermore, tests tend to show a decreasing marginal information gain through the exploration. Indeed, at the beginning only pairs of nodes are added, which creates a kind of "needles box". Then, some nodes and links are added to some pairs, which creates connected components (action movies category, as an example, becomes a connected component with more than two nodes, with five hundreds links, as shown on figure 7.1). But up to one point, a link is added which bridges the connected component to the biggest connected component. And this cannot be easily analyzed without a community detection algorithm. As an example, war movies category, which is a connected component in visualization with five hundreds links (see figure 7.1), is not a connected component anymore in the visualization with one thousand links (see figure 7.1), because it is linked to the biggest connected component and is not easily visually identifiable as a community anymore (see figure 7.1). Moreover, visualizing a bigger graph takes more time and more resources.

When analyzing the connected components representing topics from visualizations with two hundreds links (see figure 7.1), five hundreds links (see figure 7.1) and one thousand links (see figure 7.1), ones can see that most of the categories defined in section 4.3 are present as communities (more precisely as connected components) in all the different visualizations. It seems that the best visualization, which means the one which corresponds the most to the preliminary analysis with existing methods, is the one with five hundreds links. Indeed, more categories formed communities of more than two nodes (eg "action movies", "thriller movies", "horror movies" and "sport movies"), with no interesting community loss, in comparison

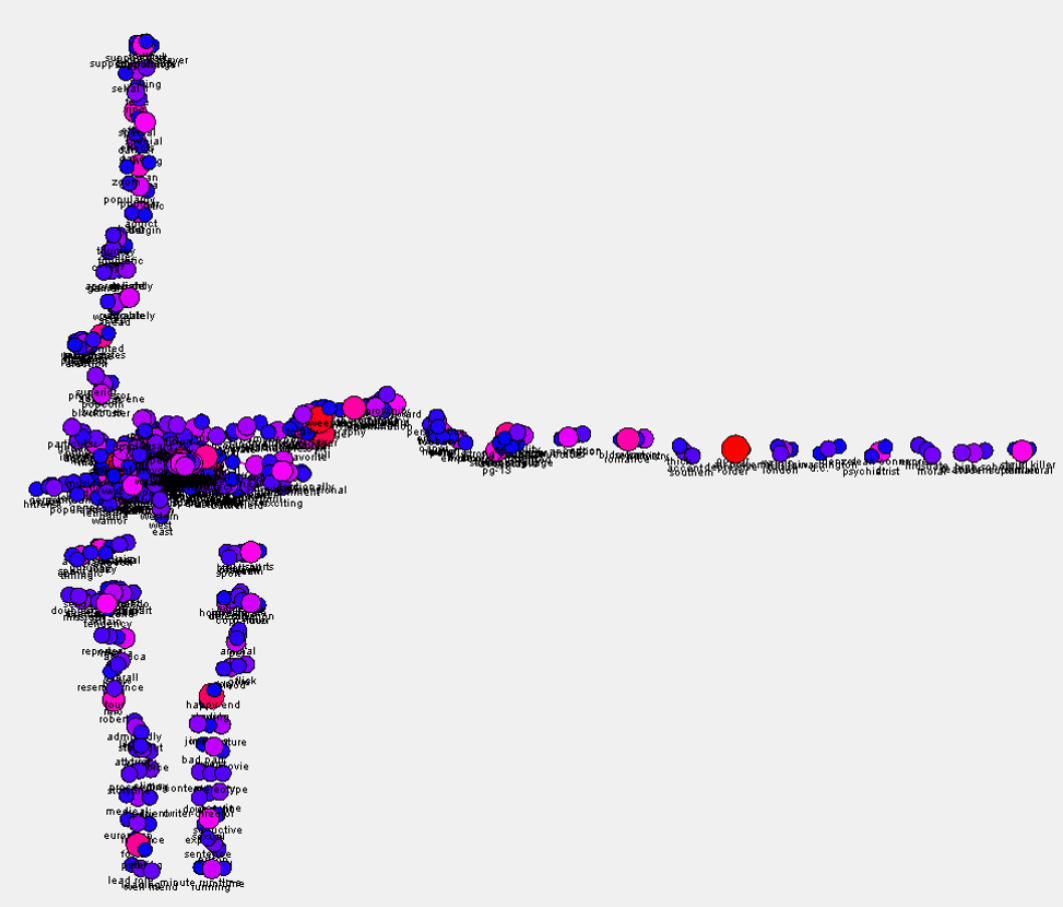


Figure 7.1: Visualization of the 1000 links with highest lift with multi-level force

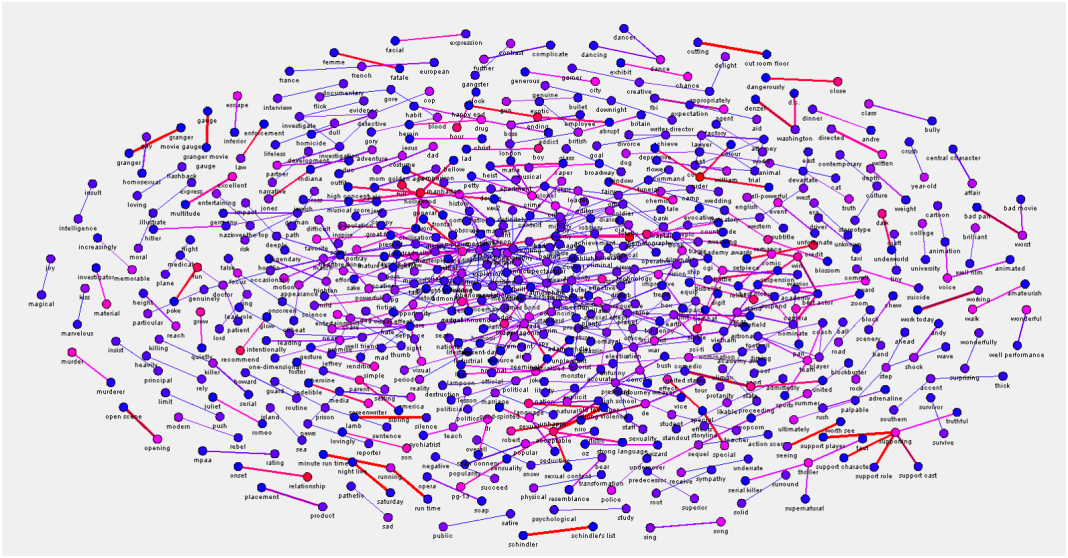


Figure 7.2: Visualization of the 1000 links with highest lift with spring force

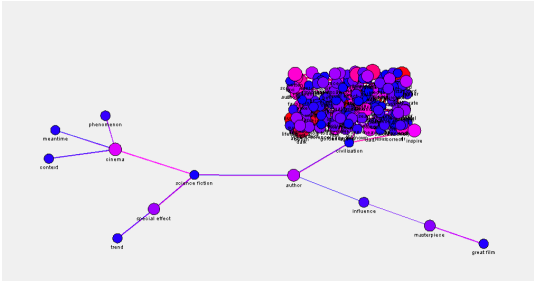


Figure 7.3: Outside the "needle box"

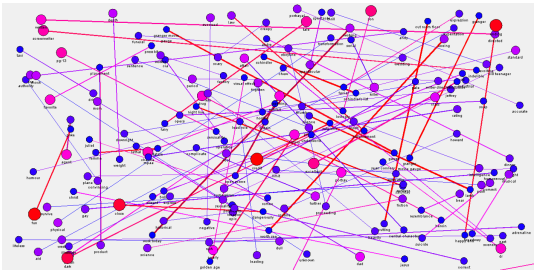


Figure 7.4: Inside the "needle box"

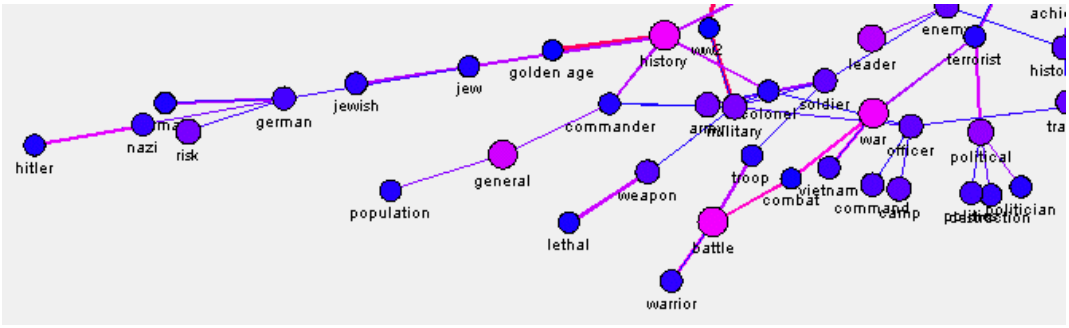


Figure 7.5: War movies theme linked to the biggest connected component

to the visualization with two hundreds links. When analyzing the one thousand links visualization it has to be noted than three important categories are joined with the main biggest component (eg "war movies", "sci-fi" and "history movie"), which makes them harder to identify, while adding no new interesting community.

However, the developed method misses some topics and includes new ones. Indeed, the three biggest topics from the preliminary analysis (see section 4.3), namely "comedy", "romantic" and "romantic comedy", are not present in the visualization. In fact, it appears that these topics exist in the developed method, but are farther in the sorted rules table (sorted by descending lift). When increasing the minimum number of documents threshold to ten, such topics are then more important and appear in the graph (see figure 7.1). In fact, it could be expected that the most frequent topics would not have a very high lift because the lift is normalized through the expected confidence (which is the probability to find a given word in the whole corpus), so the more frequent a word is, the higher the normalization factor.

Then, some small new topics are found through the analysis: "animated, movie" topic (probably linked to Disney topic), "news, media" topic, "medical, doctor, patient" topic and "dog, pet, animal" topic. These topics are very small, but very discriminative. As an example, "medical, doctor, patient" can refer to medical movies in the fashion of "ER" or "House". These very small topics could be very useful in a supervised learning task, since they help to decompose each topics into sub-topics rather than generalizing too much. Then, each movie would be described by a kind of DNA of small sub-topics (in the same way "tags" inform us of the content of a

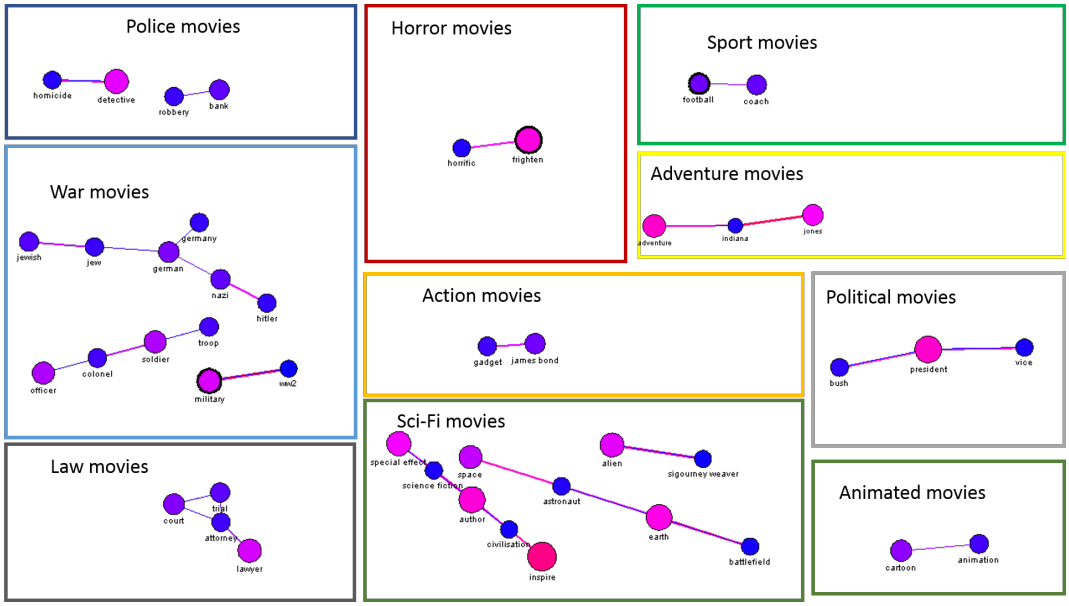


Figure 7.6: Visualization of the 200 links with highest lift with multi-level force

blog post, a web page or simply movies in streaming platforms such as Netflix.

7.2 Community detection

This section discusses the results from community detection algorithms (Louvain and Label Propagation). These results are compared with those from the visualization and the results from section 4.3.

It appeared that Label Propagation's algorithm performs better than Louvain's. Indeed, as expected, Louvain produced fewer but bigger communities (see appendix E). The fact that there are fewer communities is not an issue in itself, because there are actually fewer communities to look at. Nevertheless, the fact that communities are so big leads to mix a lot of different pieces of information. If some big topics are quite well identified ("thriller", "war", "sci-fi", "award", "dance/music", "history" and "teenager") and that it seems that community 31 represents "comedy" (which was missing in visualization), the rest is globally not interesting. There is more noise inside these clusters ("news media" topic is inside "war" topic), some communities

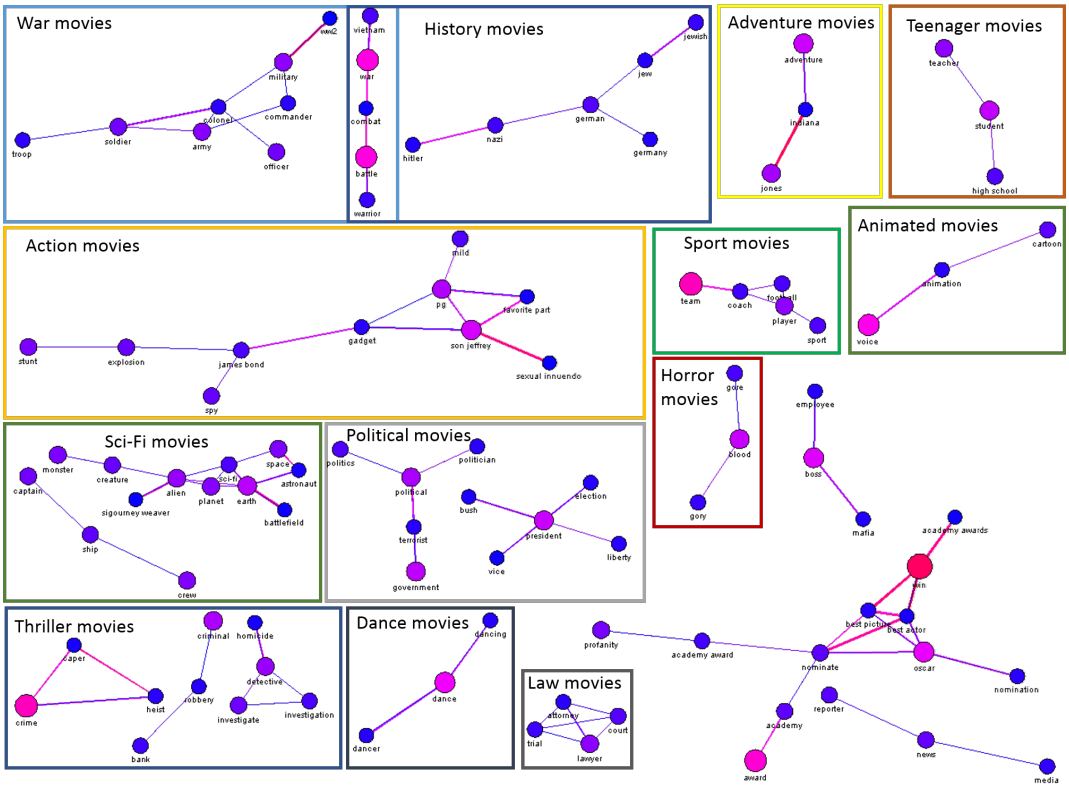


Figure 7.7: Visualization of the 500 links with highest lift with multi-level force

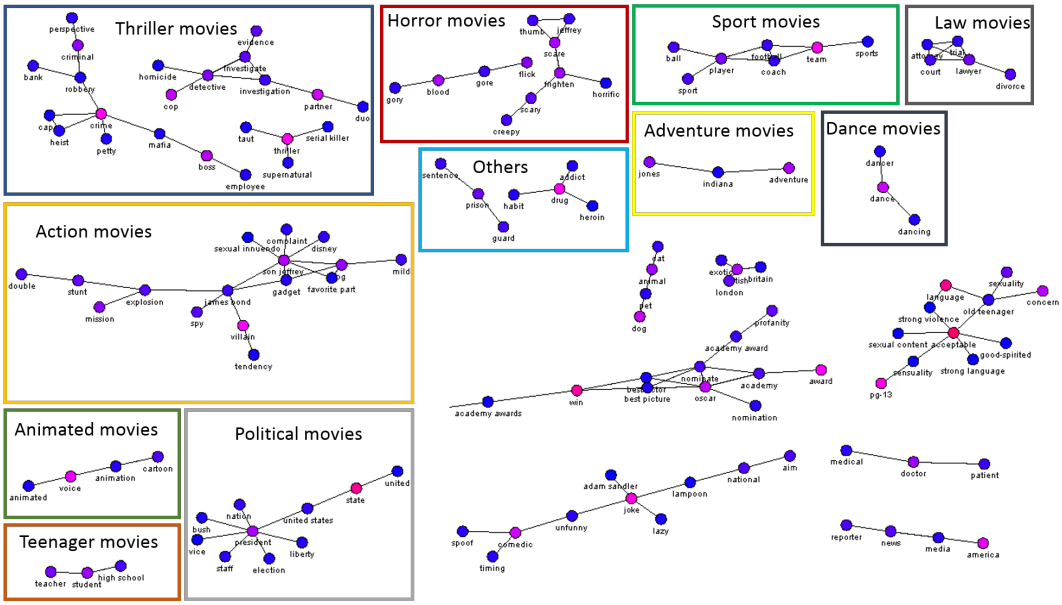


Figure 7.8: Visualization of the 1000 links with highest lift with multi-level force

seem to merge some (community 16 seems to mix "adventure movies" with "foreign movies") and some are so mixed that they did not represent an actual topic ("jesus", "christ", "taxi", "driver", "special effects", "heroin", "drug"...). Different resolution limits were tested without improving these results.

In opposition, label propagation produces mostly smaller topics which are generally more consistent. The following topics, from section 4.3, are present: "war", "sci-fi", "award", "thriller", "political", "sport", "musical", "history", "action", "law", "horror",... They are considered as present since communities from the analysis share common keywords with the analysis from section 4.3. The produced communities are also similar to the connected components that were found in the previous section. In opposition to Louvain's algorithm, Label Propagation does not produce communities of more or less the same size, so it is more flexible than Louvain's (meaning that it adapts itself better to the underlying structure of the data).

It is interesting to note that increasing the number of links used (from two thousands to height thousands) does not produce much better communities. Indeed, some small subtopics are effectively added and some topics which were previously

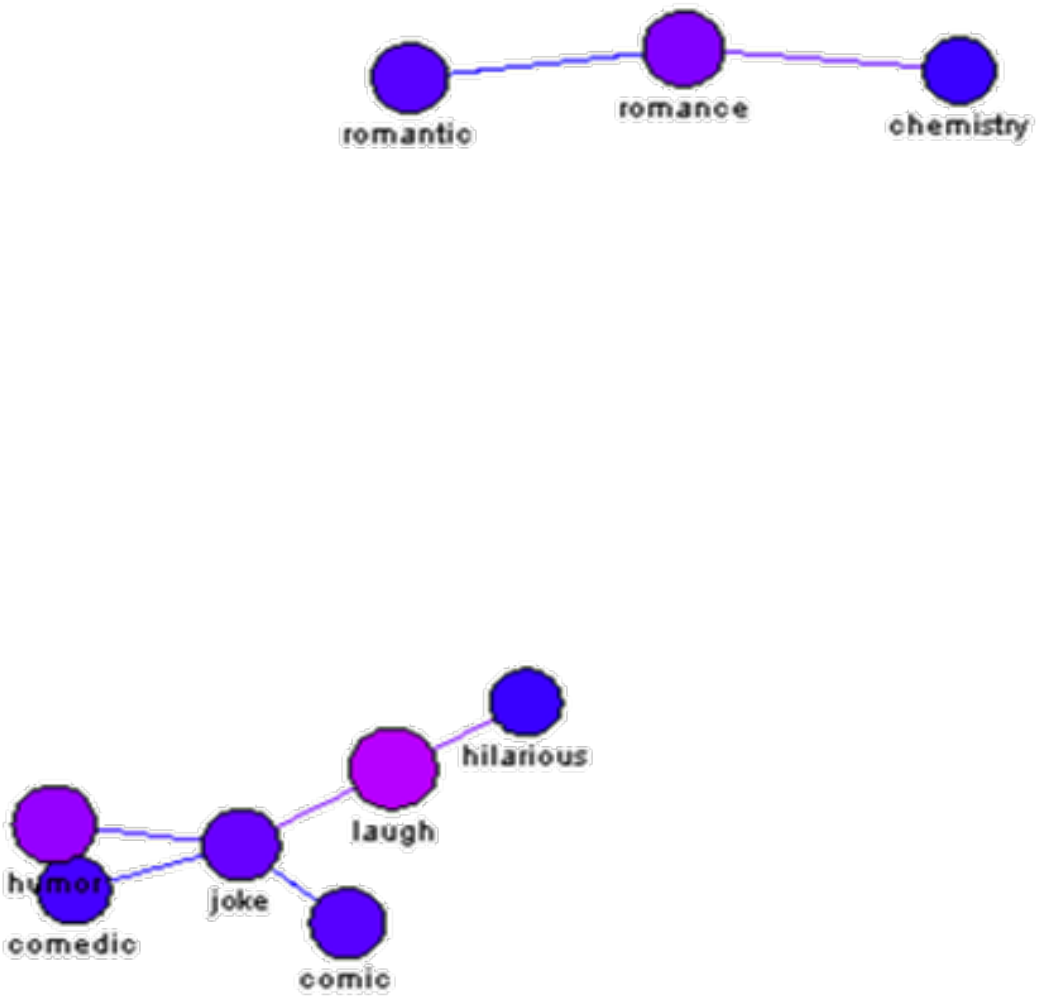


Figure 7.9: Visualization of the 20 links with highest lift with multi-level force where mindocs threshold is set at 10

spread in multiple communities are merged, but the core topics and their respective keywords are present in both cases. For example, for "war" topic (community number two for both the analysis with two thousand links and the analysis with eight thousands links), the first fourteen keywords are exactly the same. Then, it seems that when increasing the number of links analyzed Vietnam leaves the community and the small communities which were formed of "government", "official", "east" and "west" enters the community as well as the small "box" and "office" ones. But communities are globally similar.

Conclusion

In conclusion, we saw through this thesis that there exists an alternative to existing Text Analytics techniques to explore a document collection. Indeed, it appeared that the method developed performs at least as well as existing techniques and provides a new way to visualize the results of such an analysis.

We saw in chapter 5 how to use NLP to parse the textual data and filter discriminative terms. Then, in chapter 6 we saw how to use association analysis to create a graph out of these terms. Finally we have seen in chapter 7, how well the method performed (visually or not) in comparison with existing Text Mining methods.

The main advantage of this technique among others is its ability to be visualized. Indeed, we saw in section 7.1 that the network visualization is especially well-suited for visual exploration, in which communities (so the different topics) appear and catch the reader's eyes. Such an easy visualization is very appreciated by business experts because they are not necessarily technical experts, but also because it provides them with an easily understandable thing to show to the management.

Then, this method is able to extract small topics that are not extracted through regular methods. For example, one of such topics was composed of "patient", "medical" and "doctor". It can be argued that such small topics are kind of overfitting since such topics are often very specific. However, when added as tags to provide a description to each document of the collection, these small topics add useful information to differentiate between documents that belong to the same "more general" topics. For example, the series "Rosewood" would probably be classified in the general topic "thriller", it could also belong to that small "medical" topic which helps to differentiate it from another "thriller" movie.

Moreover, another advantage is its white box nature. Indeed, at the end the reason why a word is in the graph and in a given topic is clear: we know each

characteristic of a rule and its words (from the weight to the lift). In opposition, in a regular technique such as topic analysis, the whole process is blackbox (probably because it is under patent) and, although it produces good results, it does not give a clue on the validity of such an analysis. Furthermore, let us take another regular technique: the SVD. Although the computation is clear and known (we know how SVD is computed), it is considered as blackbox because the association between a word and its cluster is not clear.

Finally, another advantage of this method is that it uses existing techniques solely. So it is not necessary to develop new things, but simply to orchestrate existing ones in the manner described inside the thesis. It is possible to implement the same method in other software packages (such as Python or R) very easily now that the key elements of the implementation have been found.

The unsupervised learning task could also be improved by computing a similarity matrix based on the shared subgraphs between documents and applying a clustering algorithm to it. The more links and nodes subgraphs share, the more similar the documents are. Indeed, the graph generated with this method aims to remove non informative words (so the noise), so, by only counting the meaningful links shared by two documents, it should be possible to compute a similarity measure exempted from "common words".

Although the aim of this thesis was to explore such a method for unsupervised learning purpose, this can also be used in the future for supervised learning. Indeed, this method could be used as a dimension reduction method where each community would be a new dimension which would compute the percentage of links and nodes shared by a document with this community.

Bibliography

Albright, Russ. 2004. Taming text with the svd. *Sas institute inc., cary, nc.*

Ames, Michael. 2016. Text mining secretary clinton's emails. *In: Proceeding of the 2016 sas global forum.* Cary, NC: SAS Institute Inc. Online: http://support.sas.com/resources/papers/proceedings16/SAS6740-2016.pdf , consulted on 21/08/2017.

Ananyan, Sergei, & Kiselev, Michael. 2001. *Automated analysis of unstructured texts: Technology and implementations.* Tech. rept. Megaputer Intelligence Inc.

Azevedo, Ana Isabel Rojão Lourenço. 2008. Kdd, semma and crisp-dm: a parallel overview. *Pages 182–185 of: In proceedings iadis european conference on data mining.* IADIS.

Baesens, Bart, Van Vlasselaer, Veronique, & Verbeke, Wouter. 2015. *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection.* John Wiley & Sons.

Blondel, Vincent D, Guillaume, Jean-Loup, Lambiotte, Renaud, & Lefebvre, Etienne. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, **2008**(10), P10008.

Bouttier, Jérémie, Di Francesco, Philippe, & Guitter, Emmanuel. 2003. Geodesic distance in planar graphs. *Nuclear physics b*, **663**(3), 535–567.

Chen, Hsinchun. 2001. *Knowledge management systems.* AI Knowledge Computing Corp.

Cuvelier, Etienne, & Aufaure, Marie-Aude. 2011. Graph mining and communities detection. *ebiss*.

Duijn, Paul, & Klerks, Peter. 2014. De brug tussen wetenschap en opsporingspraktijk: Onderzoek naar de toepassing van sociale netwerkanalyse in de opsporing. *In: Tijdschrift voor criminologie*. Boom-Lemma Uitgevers.

Fan, Weiguo, Wallace, Linda, Rich, Stephanie, & Zhang, Zhongju. 2006. Tapping the power of text mining. *Communications of the acm*, **49**(September).

Feldman, Ronen, & Sanger, James. 2007. *The text mining handbook*. Cambridge University Press.

Friedman, Jerome H. 1998. Data mining and statistics: What's the connection? *Computing science and statistics*, **29**(1), 3–9.

Garnier, Alain. 2007. *L'information non structurée dans l'entreprise: usages et outils*. Lavoisier.

Ghosal, Prasun, Samanta, Tuhina, Rahaman, Hafizur, & Dasgupta, Parthasarathi. 2005. Recent trends in the application of meta-heuristics to vlsi layout design. *Pages 232–251 of: Iicai*.

Gupta, Vishal, & Lehal, Gurpreet S. 2009. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, **1**(August), 60–76.

Hossain, Muktadir, Tasnim, Tajkia, Shatabda, Swakkhar, & Farid, Dewan M. 2014. Stochastic local search for pattern set mining. *Pages 1–6 of: Software, knowledge, information management and applications (skima), 2014 8th international conference on*. IEEE.

Ibekwe-SanJuan, Fidelia. 2007. *Fouille de textes: méthodes, outils et applications*. Lavoisier.

Inc., SAS Institute. 2015. *Sas® optgraph procedure 14.1: Graph algorithms and network analysis*. Cary, NC: SAS Institute Inc.

Inc., SAS Institute. 2016. *Sas® text miner 14.2: High-performance procedures.* Cary, NC: SAS Institute Inc.

Knappen, Koen. 2015. Text analytics using sas® text miner. *In: Text analytics using sas® text miner.* Course on Text Analytics.

Lambiotte, Renaud, & Tabourier, Lionel. 2012. *Théorie des graphes.* UNamur.

Lancichinetti, Andrea, & Fortunato, Santo. 2009. Community detection algorithms: a comparative analysis. *Physical review e*, **80**(5), 056117.

Lieu, Ye, Lee, Taiyeong, Zhang, Ruiwen, & Dean, Jared. 2014. Link analysis using sas enterprise miner. *"extra" sas global forum papers.*

Mandreoli, F., Martoglia, R., & Tiberio, P. 2007. Text clustering as a mining task. *Pages 75–108 of: Text mining and its applications to intelligence, crm and knowledge management.* WITPress.

Milié-Frayling, N. 2007. Text processing and information retrieval. *Pages 1–45 of: Text mining and its applications to intelligence, crm and knowledge management.* WITPress.

Moon, Todd K. 1996. The expectation-maximization algorithm. *Ieee signal processing magazine*, **13**(6), 47–60.

Otte, Evelien, & Rousseau, Ronald. 2002. Social network analysis: a powerful strategy, also for the information sciences. *Journal of information science*, **28**(6), 441–453.

Pazienza, M.T. 2007. Information extraction and ... surroundings. *Pages 47–74 of: Text mining and its applications to intelligence, crm and knowledge management.* WITPress.

Raghavan, Usha Nandini, Albert, Réka, & Kumara, Soundar. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical review e*, **76**(3), 036106.

Ronhovde, Peter, & Nussinov, Zohar. 2010. Local resolution-limit-free potts model for community detection. *Physical review e*, **81**(4), 046114.

SAS Institute. 2011. *Applied analytics using sas® entreprise miner 12.1.* SAS Institute.

SAS Institute. 2014. *Text analytics using sas® text miner.* SAS Institute Inc.

Shannon, Claude E. 2001. A mathematical theory of communication. *Acm sigmobile mobile computing and communications review*, **5**(1), 19.

Skhiri, Sabri, & Jouili, Salim. 2012. Large graph mining: Recent developments, challenges and potential solution. *ebiss.*

Swanson, Don R. 1988. Migraine and magnesium - 11 neglected connections. *Perspectives in biology and medecine*, 526–557.

Zanasi, A. 2007. Preface. *In: Text mining and its applications to intelligence, crm and knowledge management.* WITPress.

Zhao, Yanchang. 2013. *R and data mining.* Academic Press.

Appendix A

Community detection algorithm

This appendix contains an assignment from a SNA course from Master Artificial Intelligence (KULeuven).

Comparison of SAS SNA, Gephi and Pajek for Communities detection

The aim of this project is to compare the different communities' detection algorithms implemented in SAS SNA and then, to compare SAS SNA with other packages, namely Gephi and Pajek.

The algorithms are compared based on the modularity of the results, the number of communities of the solution and their run times.

The software packages are compared in their ability to handle the dataset, the quality of the results (based on modularity and the number of communities found), the stochasticity of the results and the speed to run the analysis.

Dataset description

The social network comes from an online blogging communities where people can declare their friendships with other members. The 3.997.962 nodes represent bloggers and the 34.681.189 edges represent friendships. The dataset, which can be found in https://snap.stanford.edu/data/com-LiveJournal.html, is a flat file (with .txt extension) containing edges: each row is an edge with a source node and a destination node, separated by a tabulation. It is important to note that it is an undirected graph.

Data preparation

SAS is able to open .txt files. However, to simplify data transformations, the file has been converted to a SAS dataset and a name has been set to variables: "From" and "To". The optgraph procedure is able to handle directly edges tables: no further transformation was needed.

Gephi is able to open edges files, but it requires to have the column names "Source" and "Target". It was also needed to create a third column "Type" with each row set to "Undirected" to state that's an undirected graph. Nevertheless, Gephi wasn't able to open the file due to its size, so it has been downsized to keep only 99.999 nodes (and 2.509.640 edges). The software was able to open bigger files, but it crashed when a community detection algorithm or a non-random layout was run. It is maybe due to a Java virtual machine limitation (Gephi is developed in Java).

The method used to downsize the dataset has been to select the first 99.999 nodes of the dataset and then recover the matching edges. This method does not keep the structure of the original dataset, but it is simple and the analysis doesn't require to keep the structure of the dataset.

In opposition to previous software packages, the dataset had to be transformed before opening it in Pajek: the list of nodes must be written before the list of edges and columns must be separated by a space. Moreover, it is not allowed to have a node id of 0, which was the case of the dataset, so each ids have been indented.

Communities Detection

The whole dataset has been run on SAS and Pajek, since these software packages are able to handle it. There are three different algorithms for community detection implemented in SAS optgraph: the Louvain's algorithm, the Label Propagation algorithm and a parallelized version (under patent) of the Label Propagation algorithm.

```
proc optgraph

      data_links =  sna.LiveJournal;
      community
            algorithm        = Louvain
            out_level        = CommLevelOut
            resolution_list=1.0;
run;
```

The Louvain's algorithm takes 151.82 seconds to run and it finds 2244 communities with a modularity of 0.751 and a resolution of 1.0 (the default value). The label propagation algorithm takes 679.46 seconds; it finds 156.928 communities with a modularity of 0.690 with a resolution of 1.0 (this algorithm does not allow to change this parameter). Then, the parallel label propagation algorithm takes 32.26 seconds to find 227.845 with a modularity of 0.5449 and a resolution of 0.001 (which is the default value).

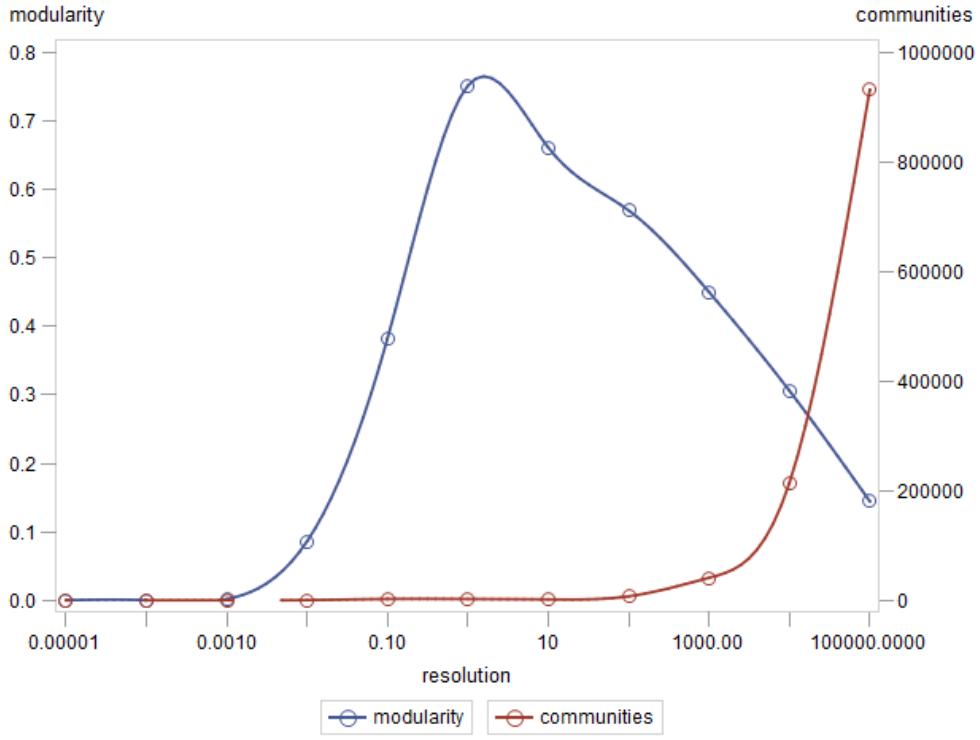
The algorithms have been run multiple time with the same parameters to check the stochasticity of the results, especially for Louvain's method which has to be compared to other packages implementation. Each run gave the same results, as explained in "SAS OPTGRAPH Procedure 14.1 Graph Algorithms and Network Analysis", the results depend on the order of the dataset. So if we would shuffle the dataset, the results would be slightly different.

It is important to note that SAS has a "tolerance" parameter so that once it is reach the algorithm stop, this parameter is explained in the paper of Blondel et al. The different algorithms have been run with default tolerance of 0.001. The parameter limiting the number of iterations had to be increase from its default value of 100 to 150 to reach convergence. Decreasing the parameter tolerance gives different results, not always better because it can be stuck in another local optimum, but it increases significantly the running time: with Louvain's algorithm, it takes 94 seconds with a 0.1 tolerance, 151 seconds with a 0.001 tolerance (which is the default value) 281 seconds for 10e-5 and 331 seconds for 10e-7.

The first observation is that Louvain's method gives much less communities as the others. Then, the second observation is that it gives a better modularity, which is expectable since this algorithm aims to maximize the modularity, while both versions of label propagation have a completely different way of working. These observations lead to one question: is it possible to find similar communities structures with the different algorithm: similar number of communities and similar modularity.

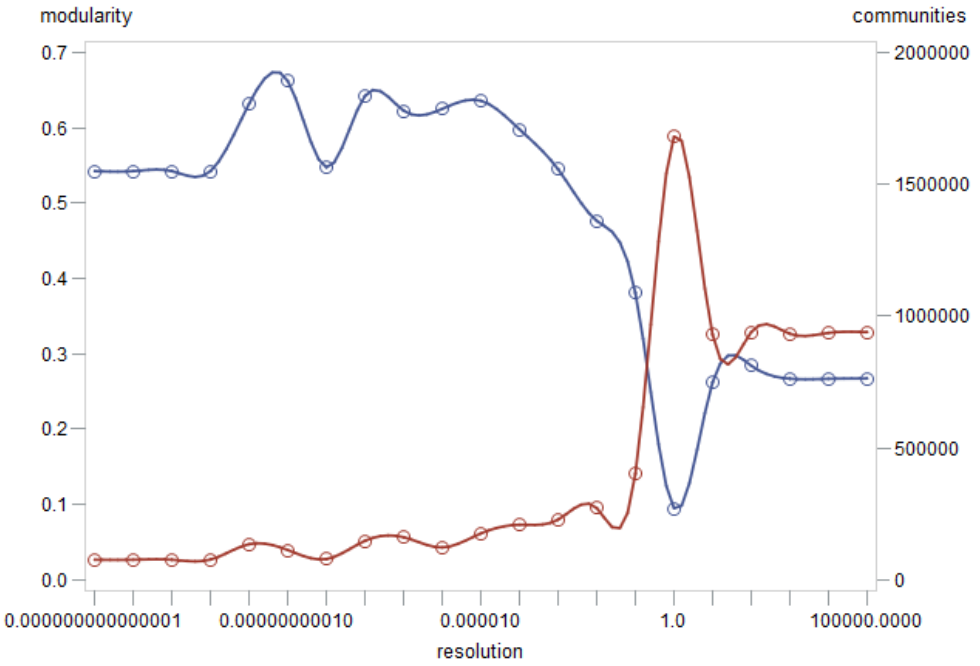
That is the reason why different values of resolutions have been tested for Louvain's method and Parallel label propagation, but also to check if the default value maximizes the modularity. It is important to note the resolution parameters of both Louvain's algorithm and Parallel Label propagation can't be compared since they don't have the same meaning, but they both aim to tune the number of communities.

For Louvain, the default resolution seems to maximize more or less the modularity. However, for parallel label propagation, it is not the case and due to the non-linearity of the results it is very difficult to find the resolution which maximizes the modularity. The same observation can be made with the small dataset.

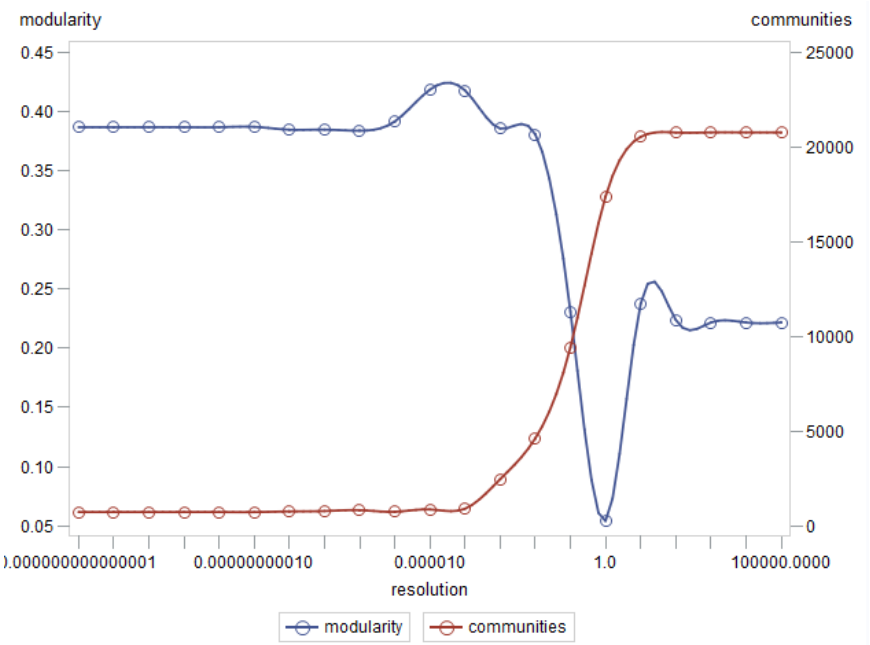


This graph shows modularity (blue) and the number of communities (red) produced by Louvain's algorithm with several resolutions (log scale)

A new observation appears regarding the resolution of parallel label propagation: from a certain resolution the number of communities as well as modularity seems to stabilize. This hypothesis has been tested with very small values: 1e-15, 1e-20, 1e-25 and 1e-100. They produce the same results 75666 communities and a modularity of 0.5425459967. This observation means it is impossible to decrease the number of communities below a certain point, while Louvain can produce solution of 1 community. It is even clearer with the small dataset.

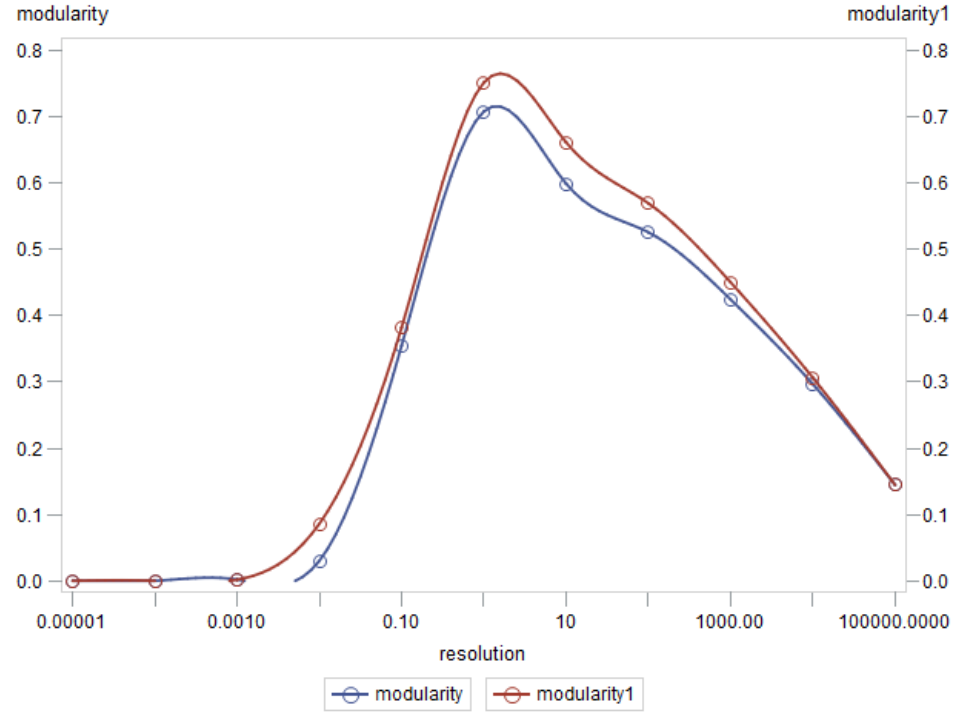


This graph shows modularity (blue) and the number of communities (red) produced by Parallel Label Propagation's algorithm with several resolutions (log scale)



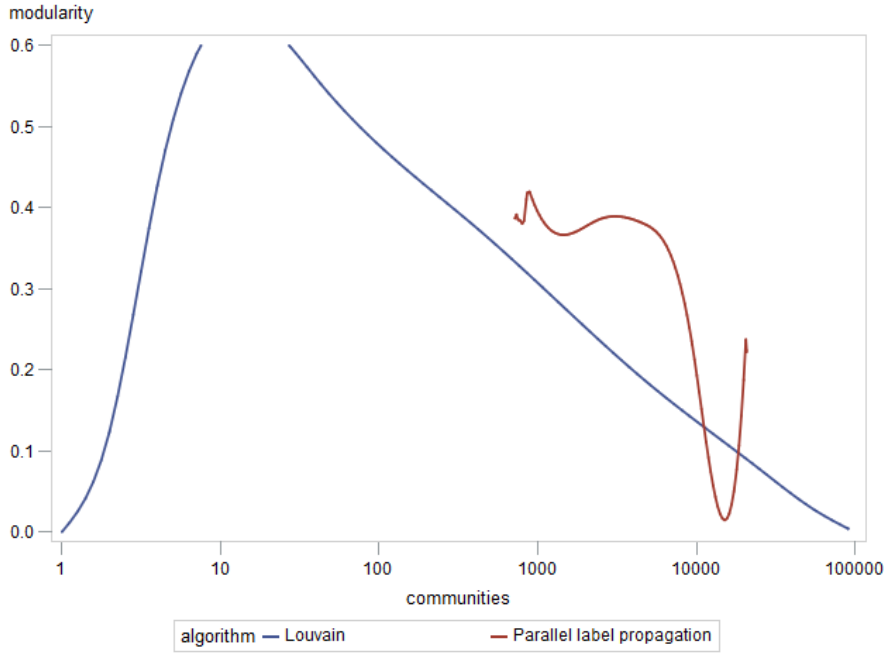
This graph shows modularity (blue) and the number of communities (red) produced by Parallel Label Propagation's algorithm with several resolutions (log scale) with the small dataset

An interesting observation is that Louvain, for a given resolution, doesn't produce the same results if this resolution is set alone or in a resolution list (unless it is the highest resolution). Indeed, the algorithm compute first the highest resolution and then merges these communities to produce results for the lower resolutions. As expected, besides the highest resolution, the results are poorer than if the resolutions would have been run apart. Nevertheless, this way of working gives an advantage over Parallel Label Propagation algorithm which restarts for each resolution: it requires only a little additional time to test several resolutions. Moreover, the "wrong modularity" seems to follow the same curve as the original one (see following graph), so it can be used to find interesting searching area. It has also been observed through experimentations with the small dataset.



This graph shows modularity produced by Louvain's algorithm, with a resolution list (blue) and without (red) with several resolutions (log scale)

The comparison of both algorithms, through a plot of modularity/communities, shows clearly that there is no clear overlap between both: Louvain doesn't manage to reach the optima found by parallel label propagation and parallel label propagation is bounded regarding the number of communities. However, there is way to escape the upper bound by using a recursive approach which is implemented. So we can conclude that the Parallel label propagation aims to find a community structure with more communities than Louvain. This has been confirmed with the small dataset.



This graph shows modularity against the number of communities produced by Louvain (blue) and Parallel Label Propagation's algorithm (red) with several resolutions (log scale) with the small dataset

This graph shows modularity against the number of communities produced by Louvain (blue) and Parallel Label Propagation's algorithm (red) with several resolutions (log scale) with the big dataset

The comparison of algorithms running times with the two datasets is consistent with what is stated in the user guide: the fact that they run in a polynomial time O(k|A|).

Comparison

Since Gephi was not able to handle the big dataset, the experiments have been run with the small dataset.

Parameters:

Randomize: Off
Use edge weights: Off
Resolution: 1.0

Results:

Modularity: 0.548
Modularity with resolution: 0.548
Number of Communities: 41

Parameters:

Randomize: On
Use edge weights: Off
Resolution: 1.0

Results:

Modularity: 0.552
Modularity with resolution: 0.552
Number of Communities: 29

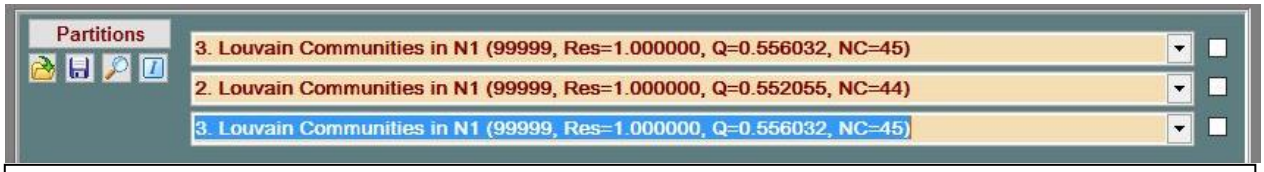
Parameters:

Randomize: Off
Use edge weights: Off
Resolution: 1.0

Results:

Modularity: 0.548
Modularity with resolution: 0.548
Number of Communities: 47

These figures show results of Louvain's algorithm with Gephi, with the small dataset



This figure shows results of Louvain's algorithm with Pajek, with the small dataset

In opposition to other packages, SAS gets always the same results, while others have a part of randomness. This is due to the sequence in which SAS runs the algorithm which is tied to the order in which edges are, rather than using a random sequence.

Then, beside the stochasticity of the search, the biggest issue to actually compare the different packages is the difference in parameters. Indeed, it is difficult to reproduce the same set of parameters since they don't propose the same parameters (eg: sas has tolerance, Gephi has Randomize parameter, Pajek has a number of random restart...).

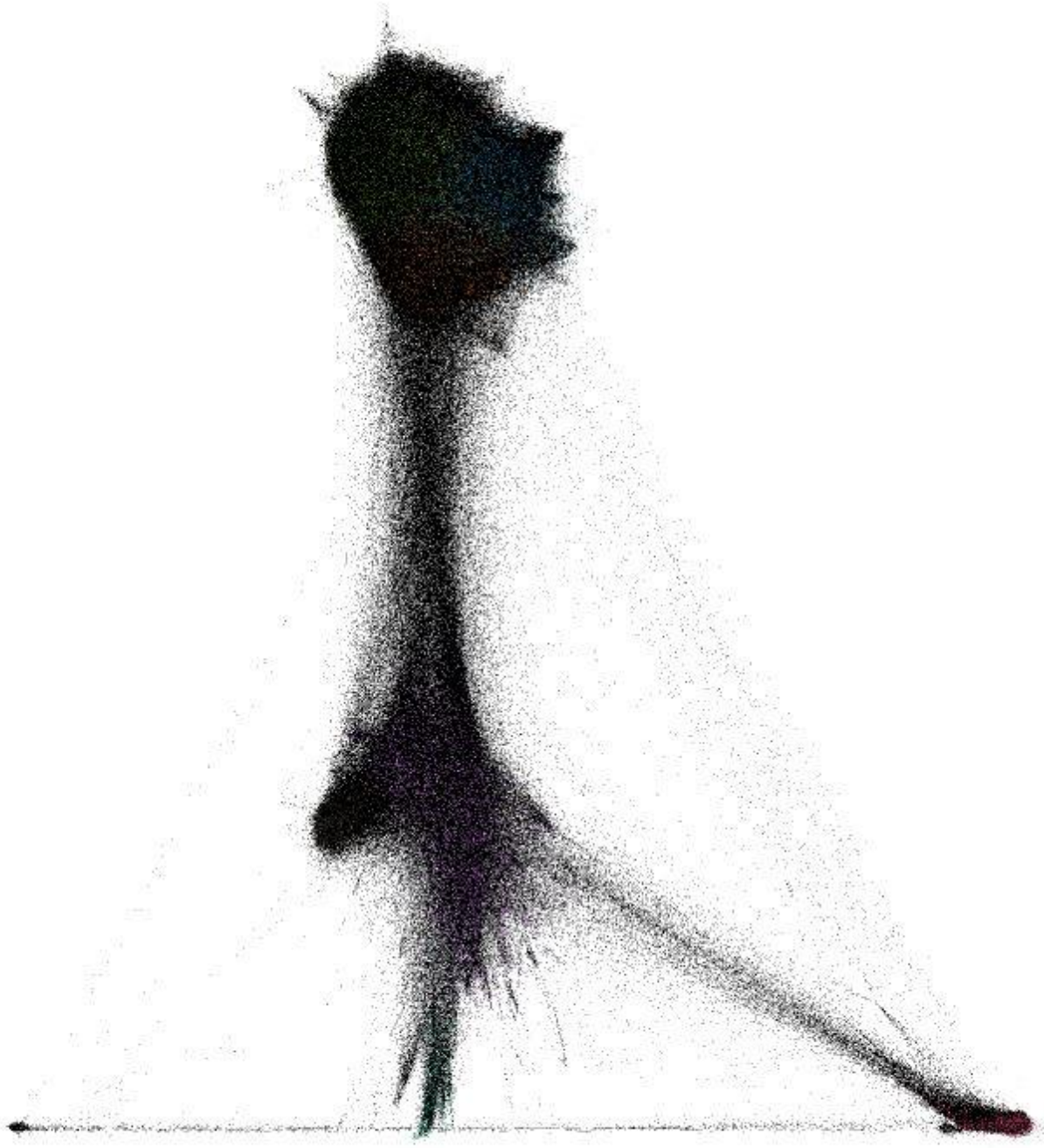
Pajek seems to produce slightly better results than Gephi and SAS but, by looking more closely to the execution log, it seems that pajek shrinks the network several times before producing the final results. If Pajek computes the modularity on the shrunk network, this couldn't be compared exactly to the results from other packages. The same observation has been done with the big dataset.

The running times are close: 8 seconds for SAS, 10 seconds for Gephi and 14 seconds for Pajek. The difference between Pajek and SAS has been confirmed with the big dataset.

Network visualization

The visualization of the network is not possible in Pajek: the software crashes quickly after the algorithm is run. The same issue occurred with SAS: the two different tools to visualize networks are not able to handle so many nodes. It mays be due to wrong installation parameters.

In Gephi we can see, with difficulties, that the communities seem consistent with the visualization.



Conclusion

The comparison of the algorithms shows there is no best algorithm. The choice of the algorithm should depend on the number of communities we want, the structure of the communities, the size of the input data, the infrastructure… If the aim is to find a lot of small communities, label propagation is a good choice. If the aim is to find the best modularity with a smaller number of communities, then Louvain is a better choice. Finally, if the aim is to decompose a huge dataset and that the server has several cores, then Parallel Label Propagation should be chosen.

The comparison of the software packages is complicated. SAS is fully scalable: it can handle huge datasets (eg someone reported an analysis with 700M nodes), then it can access any kind of data, even transactional data, it is fully integrated with other analytical products (data mining, text analytics, operational research…). However, it is not free and it requires a little SAS programming skills. In opposition, Pajek and Gephi are free and do not require any programming skills. Nevertheless, they do not have embedded data management capabilities, so the input dataset should have a predefined format. Gephi is not scalable and requires a lot of memory, but it is able to handle the small dataset for visualization while other packages are not.

Bibliography

https://snap.stanford.edu/data/com-LiveJournal.html

SAS Institute Inc. 2015. *SAS® OPTGRAPH Procedure 14.1: Graph Algorithms and Network Analysis*. Cary, NC: SAS Institute Inc.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*(3), 036106.

Appendix B

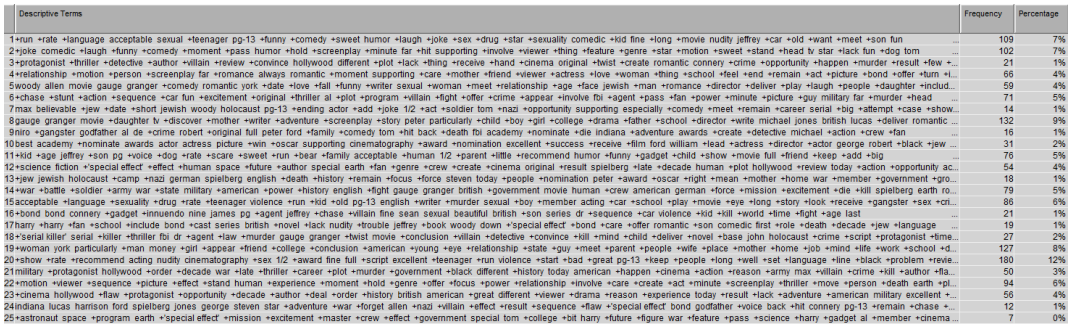
Results from cluster and topic analyses

In this chapter, the different methods that were used to extract main topics are presented. First one consists in clustering EM with the exact number of 25 topics. The second one was the entropy method with 25 topics and a min doc of 15. The third one was the same as the second one but with 40 topics. Two other tests was done : an entropy of 10 and 40 topics, for the first, and 25 topics for the second. The results was quite the same as the previous tests (entropy – min doc of 15). Thanks to these methods, different genres of movies were found. Here it is:

B.1 EM Clustering with low resolution SVD and exactly 25 clusters

This analysis was run with words in minimum 15 documents, using entropy as term weight, using SVD with 100 dimensions and low resolution, and . Results are presented in figure B.3.

- Clusters 1 and 2 : these two clusters seems to refer to comedy films. The first one is more related to sex film comedy while the second refers to comedy films in general. Indeed cluster 1 has words like "sexual", "teenager", "funny", "comedy", "sweet humor", "laugh", "joke", "sex", "drug", "sexuality comedy", "kid", "nudity" and "fun". These words could refer films like American Pie or



	Descriptive Terms	Frequency	Percentage
1	+run +rate +language acceptable sexual +teenager pg-13 +funny +comedy +sweet humor +laugh +joke +sex +drug +star +sexuality comedic +kid fine +long +movie nudity jeffrey +car +old +want +meet +son fun ...	109	7%
2	+joke comedic +laugh +funny +comedy +moment +pass humor +hold +screenplay +minute far +hit supporting +involve +viewer +thing +feature +genre +star +motion +sweet +stand +head tv star +lack fun +dog torn ...	102	7%
3	+protagonist +thriller +detective +author +villain +review +convince hollywood different +plot +lack +thing +receive +hand +cinema original +twist +create romantic connery +crime +opportunity +happen +murder +result +few +...	21	1%
4	+relationship +motion +person +screenplay far +romance always romantic +moment supporting +care +mother +friend +viewer +actress +love +woman +thing +school +feel +end +remain +act +picture +bond +offer +turn +l...	66	4%
5	+woody allen movie gauge granger +comedy romantic york +date +love +fall +funny +writer sexual +woman +meet +relationship +age +face jewish +man +romance +director +deliver +play +laugh +people +daughter +includ...	59	4%
6	+chase +stunt +action +sequence +car fun +excitement +original +thriller al +plot +program +villain +fight +offer +crime +appear +involve fbi +agent +pass +fan +power +minute +picture +guy military far +murder +head ...	71	5%
7	max believable +jew +date +short jewish woody holocaust pg-13 +ending actor +add +joke 1/2 +act +soldier tom +nazi +opportunity supporting especially +comedy +meet +remain +career serial +big +attempt +case +show...	14	1%
8	gauge granger movie +daughter tv +discover +mother +writer +adventure +screenplay +story peter particularly +child +boy +girl +college +drama +father +school +director +write michael jones british lucas +deliver romantic ...	132	9%
9	niro +gangster godfather al de +crime robert +original full peter ford +family +comedy tom +hit back +death fbi academy +nominate +die indiana +adventure awards +create +detective michael +action +crew +fan ...	16	1%
10	best academy +nominate awards actor actress picture +win +oscar supporting cinematography +award +nomination excellent +success +receive +film ford william +lead +actress +director +actor george robert +black +jew ...	31	2%
11	+kid +age jeffrey +son pg +voice +dog +rate +scare +sweet +run +bear +family acceptable +human 1/2 +parent +little +recommend humor +funny +gadget +child +show +movie full +friend +keep +add +big ...	76	5%
12	+science fiction +special effect +effect +human space +future +author special earth +fan +genre +crew +create +cinema original +result spielberg +late +decade human +plot hollywood +review today +action +opportunity ac...	54	4%
13	+jew jewish holocaust +camp +nazi german spielberg english +death +history +remain +focus +force steven today +people +nomination peter +award +oscar +right +mean +mother +home war +member +government +gro...	18	1%
14	+war +battle +soldier +army war +state military +american +power +history english +fight gauge granger british +government movie human +crew american german +force +mission +excitement +die +kill spielberg earth ro...	79	5%
15	acceptable +language +sexuality +drug +rate +teenager violence +run +kid +old pg-13 english +writer +murder sexual +boy +member acting +car +school +play +movie +eye +long +story +look +receive +gangster +sex +cri...	86	6%
16	+bond bond connery +gadget +innuendo nine james pg +agent jeffrey +chase +villain fine sean sexual beautiful british +son series dr +sequence +car violence +kid +kill +world +time +fight +age last ...	21	1%
17	harry +harry +fan +school +include bond +cast series british +novel +lack nudity +trouble jeffrey +book woody down +special effect +bond +care +offer romantic +son comedic first +role +death +decade +jew +language ...	19	1%
18	+serial killer serial +killer +thriller fbi dr +agent +law +murder gauge granger +twist movie +conclusion +villain +detective +convince +kill +mind +child +deliver +novel +base john holocaust +crime +script +protagonist +time...	27	2%
19	+woman york particularly +man money +girl +appear +friend +college +conclusion +american +young +eye +relationship +state +guy +meet +parent +people +wife +place +mother +home +job +mind +life +work +school +d...	127	8%
20	+show +rate +recommend acting nudity cinematography +sex 1/2 +award fine full +script excellent +teenager +run violence +start +bad +great pg-13 +keep +people +long +well +set +language +line +black +problem +revie...	180	12%
21	military +protagonist hollywood +order +decade war +late +thriller +career +plot +murder +government +black different +history today american +happen +cinema +action +reason +army max +villain +crime +kill +author +fla...	50	3%
22	+motion +viewer +sequence +picture +effect +stand human +experience +moment +hold +genre +offer +focus +power +relationship +involve +care +create +act +minute +screenplay +thriller +move +person +death earth +pl...	94	6%
23	+cinema hollywood +flaw +protagonist +opportunity +decade +author +deal +order +history british american +great different +viewer +drama +reason +experience today +result +lack +adventure +american military excellent +...	56	4%
24	indiana lucas harrison ford spielberg jones george steven star +adventure +war +forget allen +nazi +villain +effect +result +sequence +flaw +'special effect' bond godfather +voice back +hit connery pg-13 +remain +chase +...	12	1%
25	+astronaut space +program earth +'special effect' +mission +excitement +master +crew +effect +government special tom +college +bit harry +future +figure war +feature +pass +science +harry +gadget al +member +cinema ...	7	0%

Figure B.1: Text clustering with Movies dataset

Scary Movie and both are in the database. Cluster 2 has these words : "joke comedic", "laugh", "funny", "comedy" and "humor".

- Clusters 3 and 18 : those clusters refer to thriller category with words such as "protagonist", "thriller", "detective", "villain", "crime" and "murder" (cluster 3) ; "serial killer", "serial", "killer", "thriller", "fbi", "agent", "law", "murder", "villain", "detective", "kill" and "crime (cluster 18).

- Cluster 4 : the words in this cluster describe probably romantic movies. Indeed, there are "relationship", "romance", "romantic", "care", "friend", "actress", "love", "woman" and "date".

- Cluster 5 : this cluster refers to comedy movies but a specific one: Woody Allen's movies. Indeed, the words are : "woody", "allen", "comedy", "romantic", "date", "love", "funny", "woman", "meet", "relationship", "jewish", "man" and "romance". The most of Allen's movies are comedies and one of his recurrent character, played by himself, is an intellectual Jewish.

- Cluster 6 : this cluster describes probably action movies with words such as "chase", "stunt", "action", "car" and "fight".

- Clusters 7 and 13 : these clusters refer to war movies. Indeed, cluster 7 has these keywords : "jew", "jewish", "holocaust", "soldier" and "nazi". Cluster 13 words are : " jew", "jewish", "holocaust", "camp", "nazi", "german",

"Spielberg", "death", "history", "Steven" (we supposed that Steven refers to the firstname of Spielberg) and "war".

- Cluster 8 : this one is not related to a genre but to a situation we can notice in movies : "daughter", "mother", "child", "boy", "girl", "college", "father" and "school".

- Cluster 9 : this cluster represents action movies. Indeed the words are "gangster, "crime", "death", "fbi" and "academy".

- Cluster 10 : this one refers not to a genre but to award theme : "academy", "nominate", "awards", "win", "Oscar", "award", "nomination", "success" and "receive".

- Clusters 12 and 25 : words in cluster 12 describe Sci-Fi movies : "science", "fiction", "special effect", "effect", "human", "space", "future", "special", "earth" and "Spielberg". There, keeping "Spielberg" could be interesting because it confirms the Sci-Fi category given that Steven Spielberg produced a lot of Sci-Fi movies. Words in cluster 25 are : "astronaut", "space", "program", "earth", "special effect", "mission", "crew", "government", "future" and "gadget".

- Clusters 14 and 21 : here, keywords are related to war movies. Indeed there are "war", "battle", "soldier", "army", "war", "state", "military", "American", "history", "fight", "government", "german", "force", "mission", "die" and "kill" (cluster 14); "military", "war" and "army" (cluster 21).

- Cluster 15 :

- Clusters 16 and 17 : those clusters describe action movies. Keywords are "bond", "gadget", "james", "agent", "chase", "villain", "sexual", "british", "car", "violence" and "kill for cluster 16 and ""bond", "british", "nudity", "special effect" for cluster 17.

- Cluster 19 : it refers to romantic movies : "woman", "man", "money", "girl", "friend", "college", young", "relationship", "guy", "meet", "parent", "people", "wife", "mother", "home", "job" and "school"

Topic	Number of Terms	# Docs
max,max,gibson,marty,mel	658	129
+book,+adaptation,+century,+novel,british	691	120
+show,+recommend,acting,nudity,+sex	624	236
hollywood,+protagonist,+order,+cinema,+author	670	149
+motion,+viewer,+moment,nearly,+motion picture	765	240
+woman,york,+american,+fellow,+man	745	186
+president,+thriller,+murder,+killer,+law	648	191
+bond,bond,connery,james,+innuendo	354	54
best,+nominate,supporting,+win,+oscar	336	71
+alien,+science,alien,fiction,earth	484	137
+war,+soldier,war,+battle,+army	597	146
chan,jackie,+action,martial,kong	660	148
+comedy,+joke,sandler,+funny,+laugh	676	200
granger,gauge,movie,+daughter,+revolve	615	223
acceptable,+language,+drug,sexual,+rate	681	236
horror,+horror,+scary,+killer,+horror movie	695	171
+town,+family,+father,+son,texas	720	206
sandler,+jew,adam,+child,+parent	681	132
+school,+student,+high,+girl,football	633	199
harry,+harry,dvd,san,francisco	492	88
+song,+musical,+sing,music,+singer	652	185
+crime,+heist,+prison,tarantino,+drug	632	198
kate,+love,romantic,+woman,ryan	691	218
+kid,disney,+dog,+age,+voice	499	156
allen,woody,keaton,annie,harry	629	119

Figure B.2: Text topics with 25 topics

- Cluster 21 : beside keywords related to war movies, there is another movie genre : the thriller. Indeed, there are "thriller", "murder", villain, "crime and "kill".
- Cluster 24 : it refers to adventure movies, and probably Indiana Jones' movies. Indeed keywords are "Indiana", "lucas", "Harrison", "ford", "Spielberg", "jones", "George", "steven", "star", "war", "nazi", "effect" and "special effect".

B.2 Topic analysis with 25 topics

This analysis was run with words in minimum 15 documents, using entropy as term weight and requesting 25 topics. Results are presented in figure B.3.

- Cluster 6 : the words refer to "martial arts" movies : "chan", "action", "stunt", "Jackie" and "martial".
- Cluster 7 : this cluster describes comedy movies. Indeed words are "comedy", "joke", "funny" and "humor".

- Cluster 8: this cluster is related to action movies because keywords are "bond", "connery" and "james".

- Cluster 9 : this one has 4 words related to award theme : "best", "nominate", "win" and "Oscar".

- Cluster 10 : here, words refer to war movies : "war", "soldier", "battle" and army".

- Cluster 11 : this cluster describes Sci-Fi movies because words are "alien", "science", "fiction" and "earth"

- Cluster 13 : this one refers to teenager movies. Indeed words are "school", "student", "girl" and "college".

- Cluster 16 : this one contains two movie genres : horror films ("horror" and "scary) and thriller films ("killer" and "thriller").

- Cluster 21 and 25 : those clusters describe thriller movies ("thriller", "murder", "cop" and "killer" for cluster 21 and "crime", "tarantino" and "prison" for cluster 25).

- Cluster 22 : words of this cluster refers to musical movies : "song", "musical", "sing" and "music".

- Cluster 23 : this one describes romantic movies ("love" and "romantic").

B.3 Topic analysis with 40 topics

This analysis was run with words in minimum 15 documents, using entropy as term weight and requesting 40 topics. Results are presented in figure B.3.

- Cluster 5 : this one is related to gang movies. Indeed words are "crime", "heist", "gang", "cop" and "criminal".

- Cluster 6 : keywords refer to comedy movies : "comedy", "funny", "joke", "humor" and "laugh".

	Topic
1	+show,+recommend,acting,dialog,nudity
2	hollywood,+protagonist,+order,+cinema,+author
3	+viewer,+motion,nearly,+moment,+minute
4	granger,gauge,movie,+revolve,tv
5	+crime,+heist,+gang,+cop,+criminal
6	+comedy,+funny,+joke,humor,+laugh
7	+dog,disney,+voice,+animal,+kid
8	+bond,bond,connery,james,+innuendo
9	best,+nominate,supporting,+win,+oscar
10	+soldier,+war,+battle,war,+army
11	+alien,alien,earth,+ship,+crew
12	+woman,york,+man,principal,+people
13	+school,+student,+girl,+high,+parent
14	acceptable,+language,+rate,+drug,+teenager
15	murphy,eddie,martin,de,+sequel
16	horror,+horror,+scary,+sequel,charlie
17	harry,+harry,san,francisco,dirty
18	+family,+daughter,+mother,+child,+parent
19	+lawyer,+murder,+case,jackson,+attorney
20	+town,sheriff,texas,local,+killer
21	max,max,gibson,mad,+dog
22	+song,+musical,+sing,music,+dance
23	tarantino,sam,quentin,willis,+bride
24	allen,woody,annie,keaton,diane
25	+jew,+camp,german,+prisoner,jewish
26	+science,fiction,+machine,arnold,schwarzenegger
27	jim,+pie,american pie,american,carrey
28	chan,jackie,martial,kong,hong
29	+president,political,+government,+conspiracy,president
30	football,+team,+coach,+player,+game
31	jones,indiana,ford,smith,+sequel
32	+car,+thriller,ryan,+crash,+cop
33	+killer,serial,dr,hopkins,fbi
34	+novel,kate,+book,french,+century
35	+novel,+book,niro,howard,+picture
36	+love,romantic,+romance,+woman,+relationship
37	jack,+ship,nicholson,+crew,carrey
38	+king,+black,+white,+lord,+god
39	sandler,adam,barrymore,rob,nicholson
40	+christmas,sam,julia,roberts,santa

Figure B.3: Text topics with 40 topics

- Cluster 7 : this one describes Disney movies ("Disney", "voice", "animal" and "kid")
- Cluster 8 : this cluster is related to action movies because keywords are "bond", "connery" and "james".
- Cluster 9 : in this, this is not a movie genre but to award theme : "best", "nominate", "win" and "Oscar".
- Cluster 10 : this cluster describes war movies : "soldier", "war", "battle" and "army"
- Clusters 11 and 26 : words refer to Sci-Fi movies : "alien", "earth", "ship" and "crew" for cluster 11 and "science", "fiction", "machine", "Arnold" and "Schwarzenegger" for cluster 26. This one refers probably to Terminator movie which is a Sci-Fi movie.
- Clusters 13 : words of this cluster could refer to teenager movie : "school", "student", "girl" and "parent"
- Cluster 16 : it is related to horror movies: "horror" and "scary".
- Cluster 18 : this one refers to family movies because words are "family", "daughter", "mother", "child" and "parent".
- Cluster 19 : this cluster describes law movies. Indeed words are "lawyer", "murder", "case" and "attorney".
- Cluster 20 : it is related to crime films : "sheriff", "texas" and "killer"
- Cluster 22 : words describe musical movies. Indeed there are "song", "musical", "sing", "music" and "dance".
- Cluster 25 : this one refers probably to history movie because words are "jew", "camp", "german", "prisoner" and "jewish".
- Cluster 28 : this cluster describes "martial arts" movies because words are "chan", "Jackie", "martial", "kong" and "hong".

- Cluster 29 : words are related to politics, so it could be political movies. Indeed there are "president", "political", "government" and"conspiracy".
- Cluster 30 : this one describe sports "movies because words are "football", "team", "coach", "player" and "game".
- Cluster 32 : keywords refer to thriller movies, and maybe policy movie : "car", "thriller", "crash" and "cop".
- Cluster 33 : this one is related to policy movies with words such as "killer", "serial" and "fbi".
- Cluster 36 : this cluster has words which are related to romantic movies : "love", romantic", "romance", "woman" and "relationship".

Appendix C

Descriptive statistics on rules

Here are the descriptive statistics applied on flagged rules. Only statistics from significant variables are presented. The lift is not presented here because it is highly skewed and there are outliers which influence these statistics (except the median), so it required a percentile analysis (such as presented in section 6.2).

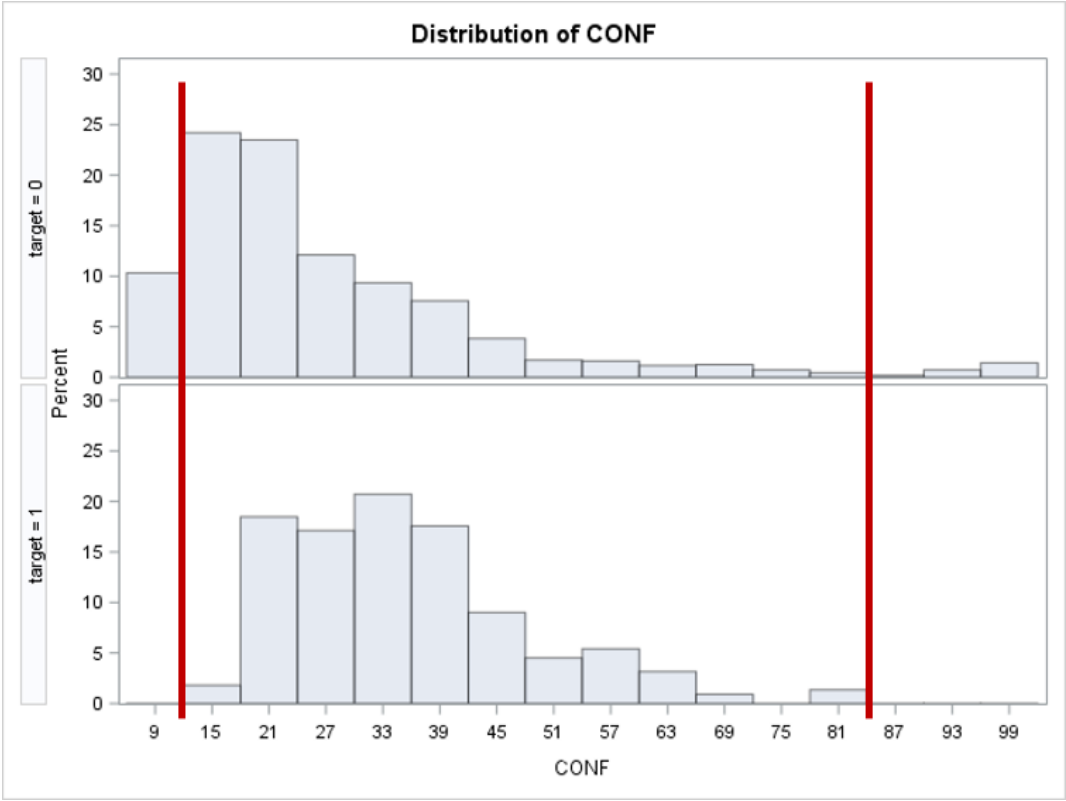


Figure C.1: Confidence distribution of meaningful rules (target=1)

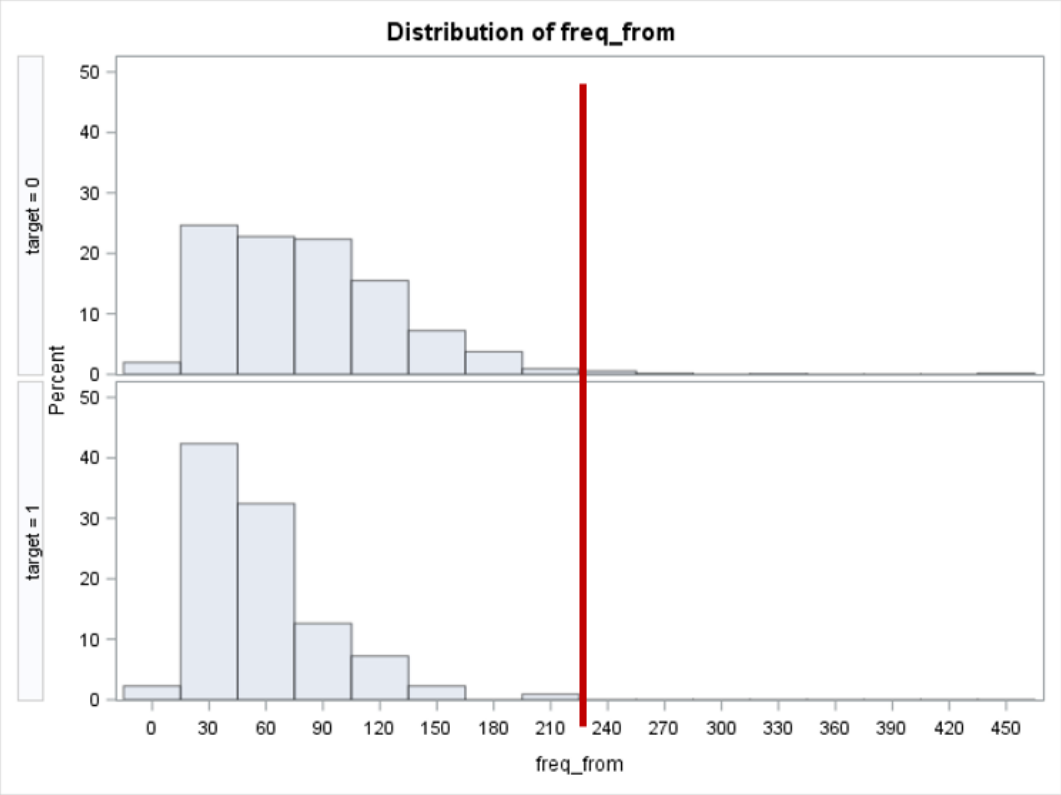


Figure C.2: Word frequency of left hand side of meaningful rules distribution (target=1)

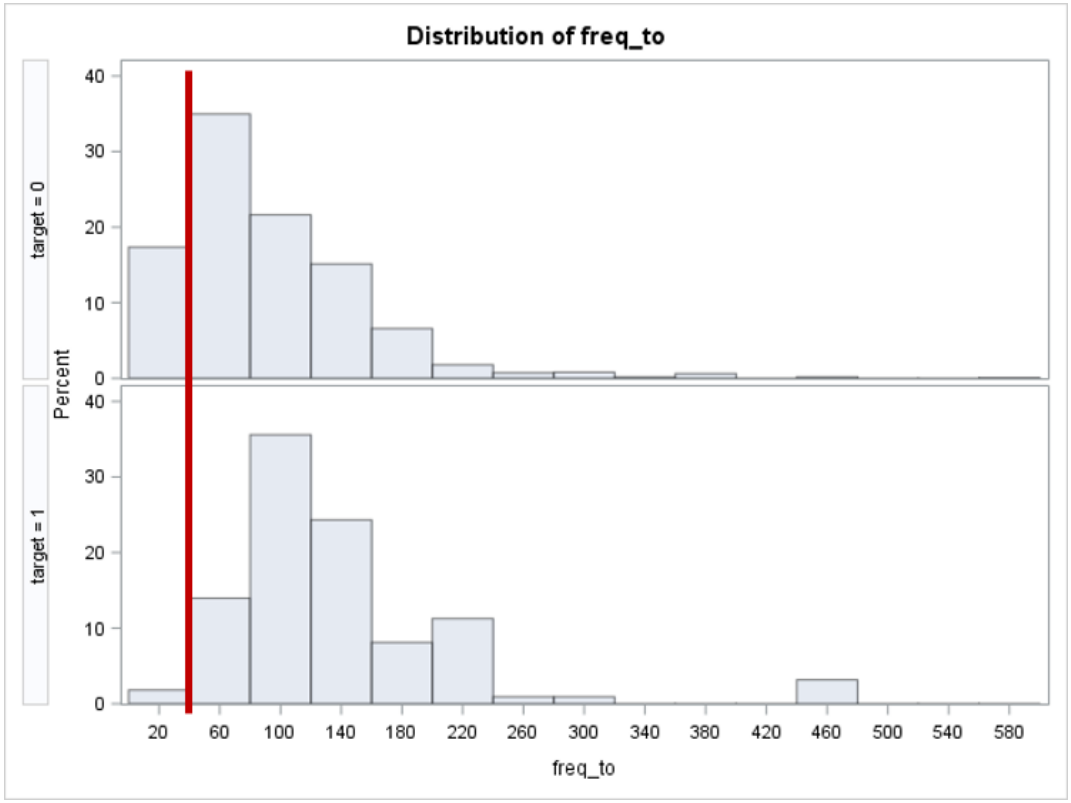


Figure C.3: Word frequency of right hand side of meaningful rules distribution (target=1)

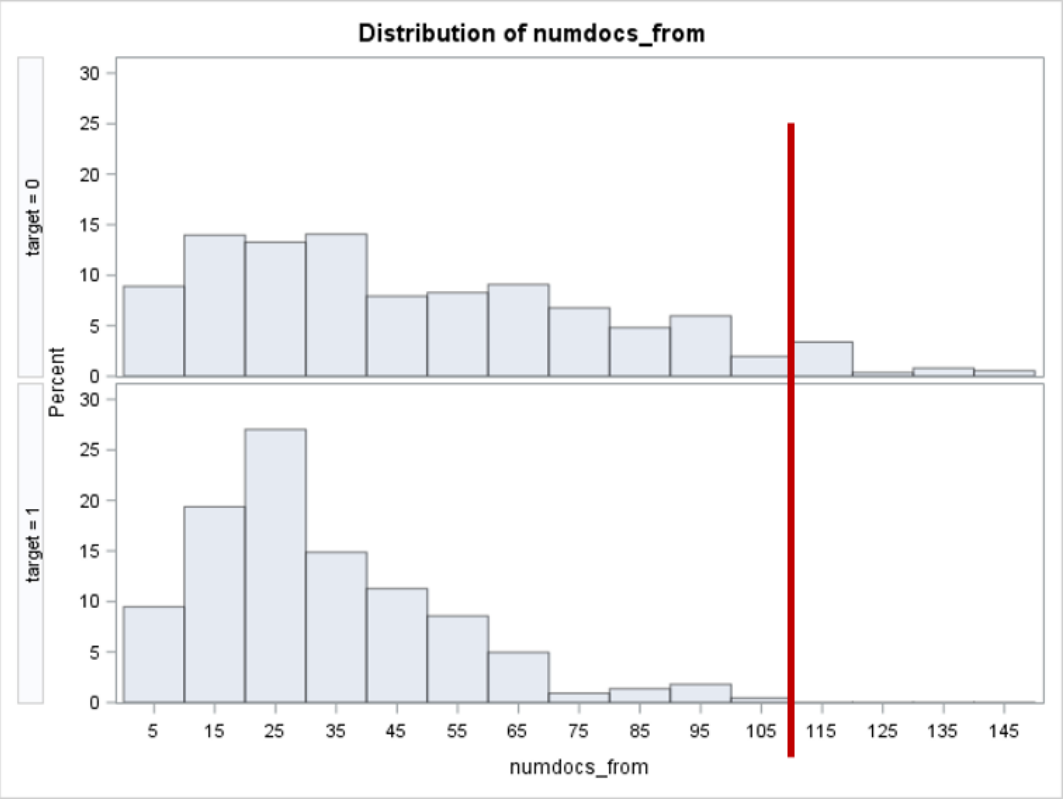


Figure C.4: Word number of documents they belong to from right hand side of meaningful rules distribution (target=1)

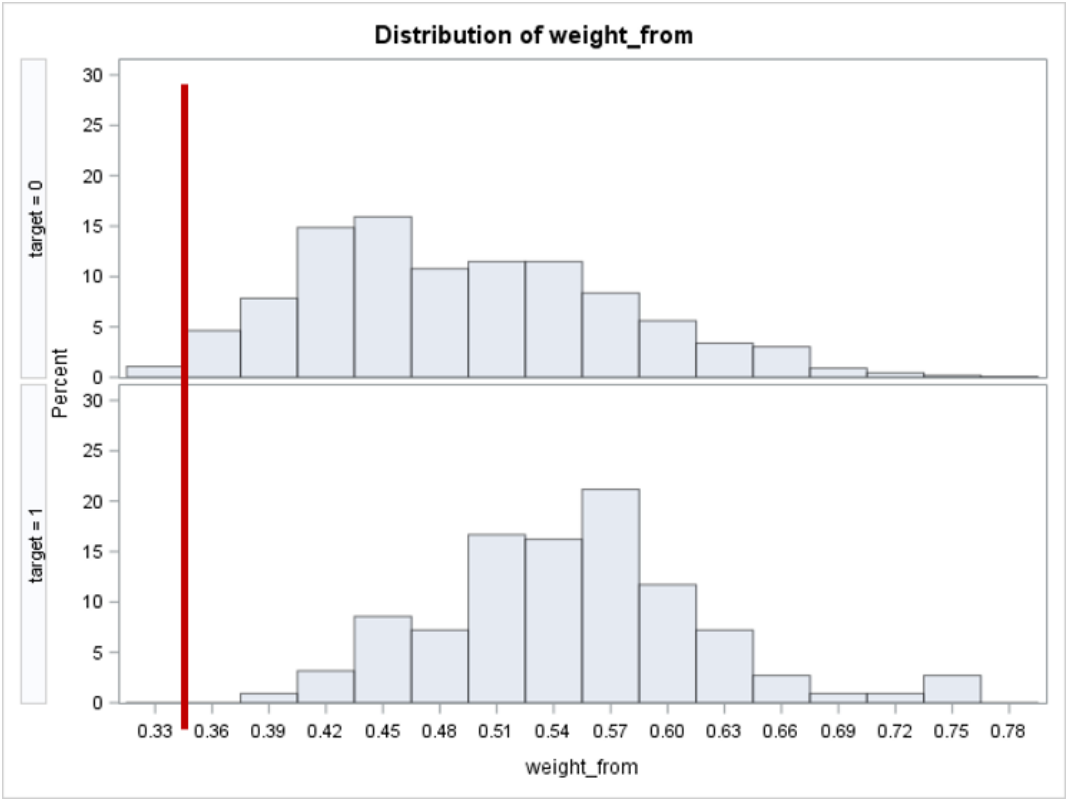


Figure C.5: Word weight of left hand side of meaningful rules distribution (target=1)

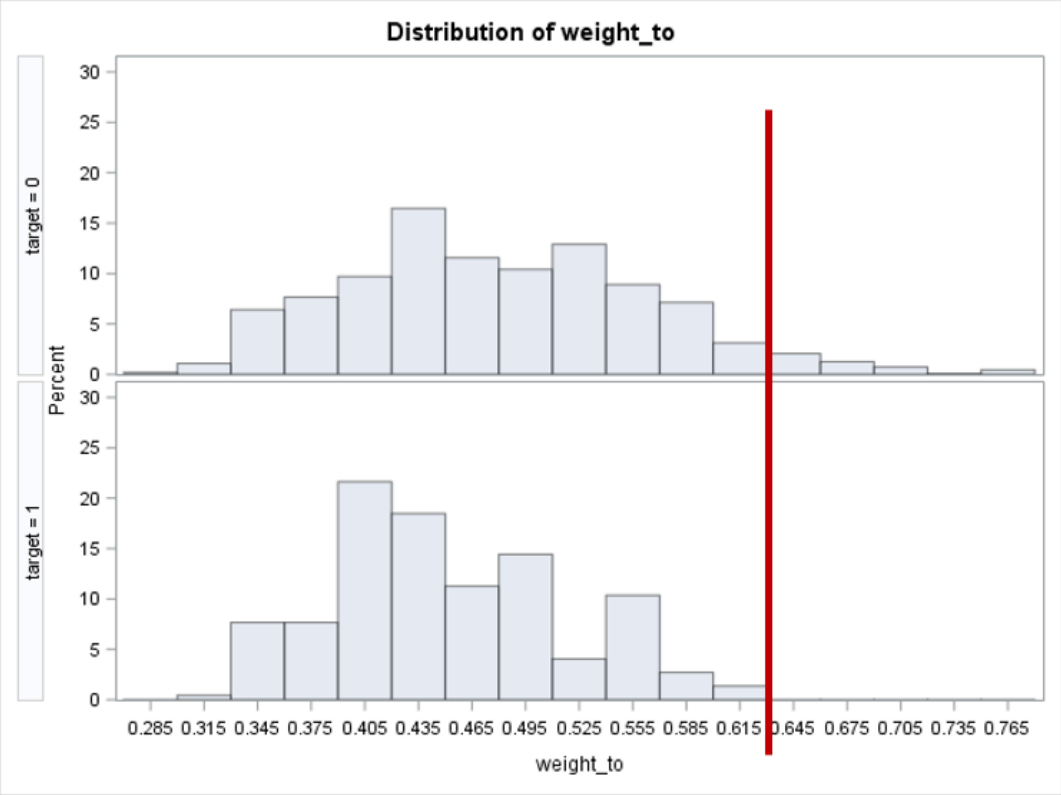


Figure C.6: Word weight of right hand side of meaningful rules distribution (target=1)

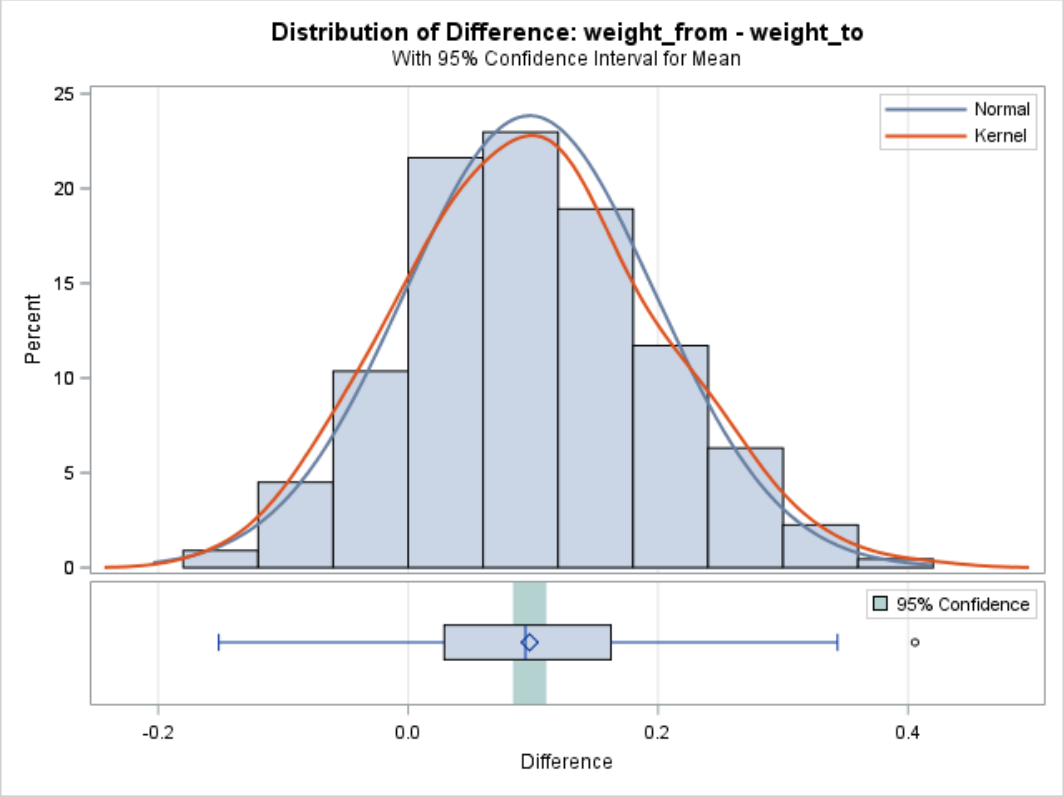


Figure C.7: Difference of both words weights of meaningful rules distribution (target=1)

target	Variable	Mean	Minimum	Maximum	Median
0	CONF	26.9874363	10	100	21.1267606
	freq_from	81.5142349	6	456	75
	freq_to	90.8478648	6	580	77
	weight_from	0.4910625	0.326346	0.7705168	0.4835201
	weight_to	0.4794825	0.2832873	0.7555934	0.4736323
	numdocs_from	47.3781139	5	148	39
	MinWeight	0.430224	0.2832873	0.6931863	0.4267952
1	CONF	35.648488	12.5	83.3333333	34.1784339
	freq_from	57.5315315	6	216	50
	freq_to	137.3288288	25	456	117
	weight_from	0.5488353	0.3830616	0.7649867	0.5467299
	weight_to	0.4516164	0.3045994	0.6296272	0.4419822
	numdocs_from	31.963964	5	104	27.5
	MinWeight	0.4434204	0.3045994	0.6222012	0.4414636

Table C.1: Example of Support, Confidence and Lift computation for rule $SavingAccount \Rightarrow Checkingaccount$

Appendix D

Full implementation

Listing D.1: Assoc and Rulegen structure

```
%let loc=LOG;
%let glob=ENTROPY;
%let minDocs=5;
%let maxDocs=200;
%let weight=0.2;
%let dropSub=T;

/* Parsing */
proc hptmine data=&EM_IMPORT_DATA;
doc_id          %EM_ID;
 variables  %EM_TEXT;
        parse           entities=std
                                stop=SASHELP.ENGSTOP
                                /*multiterm=SASHELP.ENGMULTI*/
```

Figure D.1: EMiner flow

```
                                    syn=SASHELP.ENGSYNMS
            outterms  = terms
                                    outpos = pos
            reducef=&minDocs
                                    cellwgt=log
                                    termwgt=&glob;
        select  "Aux" "Conj" "Det" "Interj" "Part" "Prep" "Pron" "Num" /ig
 run;

/* Filtering */
data terms(drop=Parent _ispar parent_id Attribute _keep);
        set terms;
        if(weight>&weight);
        if(numdocs<&maxDocs);
        /* Comment the following line if you want to keep derivatives from
        if(_ispar ^= '.');
        if(Role not in ("CURRENCY","TIME_PERIOD","TIME", "TITLE"));
run;

proc sort data=terms;
by term;
run;

/* Grouping words */
data terms2(drop= weight freq Role key sumWeight rename=(sumWeight=weight
        set terms;
by term;
retain sumFreq 0;
retain sumWeigth 0;
if(FIRST.term) then do;
        sumFreq=0;
        sumWeight=0;
end;
sumFreq+Freq;
```

```
sumWeight+(Weight*Freq);
if(LAST.term) then do;
        sumWeight=(sumWeight/sumFreq);
output;
end;
run;



proc print data=terms2(obs=1000);
run;

/* Transactionalize */
%macro transactionalize(outterm=terms, outpos=pos, transaction=DOCUMENT, parentW
proc sql noprint;
create table trans as
        %if(&childWords=T) %then %do;
                select distinct document, %if(&transaction^="document")%then %do;
                        from &outterm as t INNER JOIN &outpos as p ON t.Term = p.
                        group by &transaction, t.term
        %end;
        %if(&childWords=T and &parentWords=T)%then %do; UNION %end;
        %if(&parentWords=T) %then %do;
                select distinct document, %if(&transaction^="document")%then %do;
                        from &outterm as t INNER JOIN &outpos as p ON t.Term = p.
                        group by &transaction, t.term
        %end;
;
%if(&dropSubComp=T) %then %do;
        %clean();
%end;

%mend;
```

```
%macro clean();
proc sql noprint;
create table trans as
select * from trans
EXCEPT all corr
select TERM, DOCUMENT, count
        from trans as t1 INNER JOIN
                (select TERM as multi, DOCUMENT as docmulti
                        from trans
                        where countw(TERM)>1) as t2
        ON DOCUMENT=docmulti and findw(strip(multi), strip(Term))>0 and st
;
quit;
%mend;

%transactionalize(outterm=terms2);


proc sql;
create table terms2 as
        select * from terms2 as t INNER JOIN(
select distinct term, document, count(*) as num_docs
        from trans) as c ON t.term=c.term
;

quit;



/* Export Transaction dataset*/
data &EM_EXPORT_TRANSACTION;
        set trans;
run;
```

```
data &EM_EXPORT_TRAIN;
        set terms2;
run;

%EM_METACHANGE(name=DOCUMENT, role=ID,key=CDELTA_TRANSACTION);
%EM_METACHANGE(name=TERM, role=TARGET,key=CDELTA_TRANSACTION);
```

Listing D.2: Assoc and Rulegen structure

```
%let support=10;
%let conf=10;
%let items=2;
%let datrule=datrule;
%let minLift=2.0;
%let keep=10000;


proc dmdb batch data=&EM_IMPORT_TRANSACTION dmdbcat=catassoc;
        id DOCUMENT ;
        class TERM(desc);
        target TERM;
run;

proc assoc data=&EM_IMPORT_TRANSACTION dmdbcat=catassoc out=datassoc(label='Outpu
cust DOCUMENT;
target TERM;
run;

proc rulegen out=&datrule(label='Output from Proc Rulegen')
minconf=&conf;
run;

data edges(drop=SET_SIZE ITEM1 ITEM2 ITEM3 rename=(_LHAND=From _RHAND=To RULE=lin
        set &datrule;
        if(countw(_RHAND)>0);
```

```
        if(lift>&minLift);
        if(_RHAND>_LHAND) then id=cats(_RHAND,'_',_LHAND);
        else id=cats(_LHAND,'_',_RHAND);
run;

proc print data=edges;
where strip(from)="joke" or strip(to)="joke";
run;

proc sort data=edges;
 by descending lift;
run;

data edges;
        set edges(obs=&keep);
run;

proc sql;

        create table tempedges as
        select To, From, count, support, lift, conf, exp_conf, weight as w
        from edges as e INNER JOIN &EM_IMPORT_DATA as t ON strip(From)=str
        ;
        create table edgesOut as
        select To, From, count, support, lift, conf, exp_conf, weight_from
        from tempedges as e INNER JOIN &EM_IMPORT_DATA as t ON strip(To)=s
        ;

quit;

data edgesOut;
        set edgesOut;
        if(freq_from<225);
        if(numdocs_from<110);
```

```
        if(weight_from>0.375);
        if(CONF>12);
        if(CONF<84);
        if(freq_to>25);
        if(weight_to<0.63);
run;

proc sql;
        create table memrules as

                select From as _LHAND from edgesOut
                UNION all corr
                select To as _LHAND from edgesOut
;
        create table nodes as
                select _LHAND, count from &datrule
                        where (countw(_RHAND)=0)
        INTERSECT all corr
                select distinct _LHAND from memrules
        ;
        create table nodes as
                select d._LHAND, count from &datrule as d INNER JOIN nodes as n C
                        where (countw(d._RHAND)=0)
        ;

        create table countmemrules as
        select distinct _LHAND, count(*) as count from memrules
        group by _LHAND
        ;

quit;


data nodesOut(keep=_LHAND COUNT rename=(_LHAND=item));
```

```
        set nodes;
run;


proc print data=edgesout;
run;

proc optgraph
        data_links = edges(rename=(lift=weight))  out_nodes=OutNodesLP;
        community
                algorithm=parallel_label_prop
                out_level        = CommLevelOutLP
                out_community= CommOutLP

        resolution_list=1E-1
;
run;

proc sort data=OutNodesLP;
by community_1;
run;

data wordNetTopics(drop=node);
        set OutNodesLP;
        by community_1;
        length topic \$ 32000;
        retain topic;
        retain nTerms;
        if(FIRST.community_1) then do;
                nTerms=1;
                topic=node;
        end;
        else do;
        topic=catx(',',topic, strip(node));
```

```
        nTerms+1;
        end;
        if(LAST.community_1) then output;
run;
proc sort data=wordNetTopics;
by descending nTerms;
run;
```

Appendix E

Community detection results

In this section, communities of word formed by the different community detection algorithms are presented. Only the first words were kept so that it fits the page.

E.1 Louvain with 8000 best lift links

E.2 Label Propagation 8000 best lift links

Community	Members
18	chew, scenery, fiction, science, science fiction, special effect, specialised
10	cia, operative, ww2, military, weapon, lethal, colonel, soldier, troop, officer
22	adrenaline, rush, heartbreaking, deeply, gradually, occasionally, difficult
23	andre, dinner, jeffrey, thumb, bad language, editor, scare, dialog, totally, sex
21	homicide, detective, commit, suicide, investigation, investigate, duo, partner
25	sexual innuendo, son jeffrey, favorite part, pg, mild, complaint, size, table
51	adam sandler, joke, unfunny, comedic, lazy, timing, spoof, fail, raunchy
0	best actor, best picture, nominate, academy award, nomination, oscar
11	james bond, gadget, spy, stunt, explosion, tendency, villain, legendary
8	d.c., washington, bush, president, vice, election, liberty, staff, nation
2	caper, heist, crime, petty, plan, crook, thief, steal, winter, deliver, convict
19	football, coach, player, team, sport, sports, ball, league, field, game, recruit
55	depth, weight, manhattan, apartment, mainstream, fault, pose, making
162	awe-inspire, sequence, alternate, sumptuous, render, map, wasted

4	jew, jewish, hitler, nazi, germany, german, risk, flat, success, critical
36	sequel, predecessor, popcorn, action scene, heroine, shining, original
54	creative, appropriately, delight, match, garner, narrative, development
191	doom, adaptation, cinematic, patrick, wise, novel, adapt, traditional
16	attorney, trial, court, lawyer, divorce, justice, law, judge, quarter, explain
17	sigourney weaver, alien, creature, monster, scientist, flick, imaginative
20	running, run time, final, consistently, thin, slight, scale, stretch, conclude
31	set-piece, giant, visual, entertainment, imagination, attack, value, brian
74	storyline, proceeding, climax, setting, loose, unlikely, threaten, date
82	thrill, exciting, frame, ride, speed, hide, clear, technical, naturally, entertain
87	likable, standout, admittedly, smart, newcomer, shallow, mess, existence
88	sibling, parent, foster, horrible, coming-of-age, intimate, hard, close-up
127	wonderfully, surprising, delightful, dress, screening, clothes, smile, sweet
143	overshadow, deal, embarrassing, increase, link, guilt, bitter, religious
147	motion picture, apparent, artist, conventional, equal, subtle, mistake, dig
163	locale, fine, stirring, plant, ian, overdo, donald, carol, sweetness, demeanor
282	prologue, return, experience, relax, understandable, territory, comfort
50	dancing, dance, dancer, shoe, folk, top, hear, explode, immediately
58	mature teenager, nudity, frontal, admonish, porn, glad, overact, suffice, devil
78	intelligence, insult, designer, production, design, insight, virtually, glimpse
94	rebel, modern, flawless, invasion, remain, reward, storytelling, ruin, stake
97	stereotype, culture, america, contemporary, explore, prime, forget, native
107	loving, flashback, childhood, tragedy, pain, desperate, memory, mention
130	gesture, simple, classic, pilot, glorious, harmless, princess, reduce, instant
133	resource, multitude, depressive, order, colour, star war, bellow, hype
145	strip, actual, doubt, gain, center, blame, determine, estrange, treat
149	cgi, comrade, lack, lower, street, sorely, uninteresting, uninspired, conviction
184	great film, allow, ill, indicate, establishment, gather, maintain, resident
224	truck, drive, crazy, sign, report, wild, luck, mile, car
26	cartoon, animation, voice, animated, disney, adult, radio, talking
28	musical, broadway, sing, song, singer, burst, shy, stone
30	plane, flight, indiana, adventure, numerous, jim, trip, master
40	terrorist, political, politician, politics, destruction, freedom, corrupt, chief
61	computer, generate, program, machine, impossible, combine, building

63	tour, de, rent, party, core, note, evoke, village
77	bully, class, student, high school, teacher, principal, insist, demonstrate
85	explicit, sexual, seductive, male, female, sexy, section, wit
86	gun, bullet, shooting, protect, grown, die, operation, accidentally
91	relief, comic, one-liner, throwaway, broad, highlight, pal, gift
160	lighting, image, remark, surreal, mental, announce, stay, harsh
166	usa, american, victory, succumb, missile, lt, japanese, foreign
168	infinitely, far, assured, big-screen, decidedly, screen time, topic, coast
180	artistic, maker, universal, important, importance, silence, category, kind
196	faithful, share, dysfunctional, ironically, depression, clueless, dealing, eddie
206	conscience, hero, code, break, menace, appealing, interest, bind
226	double, thought, knowing, sit, intelligent, incredibly, slightly, twist
312	smite, fall, centerpiece, misguide, asleep, pile, accomplished, despicable
15	astronaut, space, battlefield, earth, sci-fi, planet, blend
32	pale, comparison, remake, exchange, potential, exception, otherwise
45	worker, factory, less, participant, karen, inevitably, spring
92	outfit, costume, estate, outlandish, colorful, usual, remarkably
95	premise, intentionally, one-dimensional, creation, advantage, potentially
102	assassin, serial killer, thriller, taut, supernatural, motivate, enjoy
104	well performance, wonderful, lush, big screen, theater, convey, terrific
110	marvelous, kiss, ticket, buy, beach, list, living
111	sad, pathetic, mainly, uneven, worse, segment, suppose
142	racist, white, racism, print, racial, trite, black
152	homage, pay, price, worth, profession, buck, occasional
170	furious, fast, bound, light, criticism, fix, intend
187	caricature, cheap, max, mediocre, wish, surprisingly, extra
189	gangster, heavy, business, cash, unusual, cynical, reaction
201	annoy, present, experiment, confront, support role, multiple, industry
204	initial, new york, hall, avoid, girlfriend, lisa, innocence
210	sight, journey, reminiscent, speech, innocent, thomas, understand
214	ridiculous, awful, parody, producer, suggest, foot, promise
220	dimension, revelation, feat, person, meeting, understanding, birth
231	condition, rich, brown, heat, invite, house, intent
257	scope, hold, portrait, pointless, excite, hostage, annie

293	civil war, force, showdown, blade, heroic, gap, nuclear
12	placement, product, hell, god, absolutely, require
13	taxi, driver, exhibit, chance, closely, nurse
24	resemblance, bear, snow, overall, la, wait
34	british, britain, exotic, london, advance, martin
47	visual effect, nearly, counterpart, closer, stock, phone
53	downright, writer-director, genuine, successful, support, method
76	sympathy, root, sympathetic, struggle, debut, desire
89	love story, chemistry, romantic comedy, deny, romantic, page
99	amateurish, walk, window, glass, nose, neighborhood
109	truthful, supporting, exact, type, new york city, chicago
117	ahead, step, wave, palpable, teenage, charming
123	romance, blossom, promising, spark, solely, destine
125	university, college, graduate, professor, wrong, challenge
131	sea, island, straight, peter, miller, joe
138	suspense, tension, build, intense, cinematographer, prepare
146	yes, amaze, make-up, favor, degree, discuss
171	brilliantly, join, psychotic, true, convention, safety
173	nowhere, move, rhythm, overlook, drift, wildly
175	flow, fantasy, lonely, dream, eventually, energy
176	classmate, school, attend, act, small town, uncle
181	admirable, hit, mixed, trio, boys, balance
192	increasingly, practically, major, discover, vulnerability, sudden
209	uproarious, laugh, dave, bust, sight gag, ridicule
212	sharp, identity, witness, free, term, spot
219	concentrate, beginning, gordon, sink, formula, edward
225	concoct, realize, desperation, offer, incomprehensible, substance
227	instance, shot, carter, basis, entirely, restaurant
241	bog, down, molly, crawl, exit, exhaust
248	lifestyle, early, runner, mel gibson, governor, controversial
260	occasion, robert, embody, perfect, narration, attraction
288	fighting, fight, fighter, cheesy, companion, chinese
294	helpless, face, dilemma, label, clock, wisdom
304	bowl, mean, wipe, seriousness, cartoonish, training

323	grace, clean, honor, easily, plausible, troubled
324	metaphor, husband, photographer, stun, choose, abandon
327	loosely, base, act talent, passionate, true story, nicholas
351	half-hour, minute, inconsequential, funny moment, requirement
5	expression, facial, arrive, rarely, quirky
38	lead role, leading, well friend, sleep, boyfriend
39	fbi, agent, brain, bite, check
43	retort, hate, rendition, version, range
48	lifeless, dull, surround, solid, writing
52	heroin, drug, addict, habit, abuse
56	complicate, further, jr, pair, buddy
65	scary, creepy, frank, bomb, handle
68	sentence, prison, guard, barely, fly
70	blood, gory, limit, push, gore
72	underworld, dark, exceptional, horrify, bleak
73	transformation, physical, content, track, unable
83	succeed, negative, title, style, sun
84	cat, animal, pet, dog, beloved
96	edge, seat, green, grant, blue
98	accurate, natural, empty, conversation, adam
101	zoom, camera, pan, angle, shade
112	digit, single, awe, endure, construct
115	unit, impressive, mass, similarity, capable
119	summer, blockbuster, season, fare, smith
120	onscreen, focus, shift, grand, addition
122	inferior, escape, attractive, prisoner, border
126	unhappy, marriage, married, wealthy, passion
134	well film, brilliant, disaster, depiction, incredible
135	satire, public, target, current, perfectly
137	investigator, material, distance, worthwhile, mansion
139	realism, bond, tear, poignant, sam
140	client, company, executive, notion, exactly
148	puzzle, piece, limited, realistic, hurt
151	likewise, refuse, save, harry, mary

153	competition, perform, young man, wind, familiar
155	authority, embrace, serve, creator, sidekick
156	previous, complex, manner, issue, excuse
158	trademark, theme, wing, human being, regular
164	palace, view, lieutenant, sacrifice, shatter
167	definitely, tune, pop, accept, main
169	enhance, slow, impress, wrap, drag
174	composer, quality, anymore, desperately, courage
183	saw, first time, touch, hang, trailer
188	closing, credit, las vegas, coworker, wheel
190	theatrical, release, faith, haunt, notably
195	entry, fellow, inhabit, ironic, eager
200	tired, matter, co-star, statement, painful
203	reply, busy, wear, hat, gary
213	subplot, belief, jack, forgettable, trap
215	revolve, granger movie gauge, hip, cleverly, dump
218	central, south, ensemble, park, roll
223	unexpected, latter, legend, difference, resemble
229	power, landscape, recall, cliche, melodrama
235	george, moving, cliched, middle, bob
244	shame, fresh, half, quickly, fairly
245	assemble, follow, rough, easy, thread
249	answer, question, grave, jonathan, tree
255	first half, second, driving, subtly, burden
258	defense, large, suit, well scene, ice
261	bother, trait, situation, nearby, melodramatic
264	suddenly, tie, normal, raise, concept
277	outside, opposite, slowly, terrible, teen
279	profanity, constant, bed, worry, jump
280	afford, conclusion, physically, pass, possibility
283	score, triumph, emerge, continue, mentor
292	peace, side, brave, positive, hope
295	control, declare, wide, system, flash
300	entertaining, hole, clever, enjoyable, cameo

302	incendiary, care, cup, understated, purely
306	tightly, night, terror, cope, grip
311	afterall, kid, reassure, grin, adorable
314	suitable, reason, comic relief, compromise, flesh
319	tale, morality, compel, distant, walter
330	fortunate, place, down-to-earth, point of view, claustrophobic
338	world, slim, homicidal, seventh, dedicated
341	cause, pressure, exist, duty, strain
354	heel, one-note, head, shoulder, pump
9	politically, correct, deliberately, pace
33	sexuality, old teenager, concern, neighbor
35	thick, accent, southern, fashion
44	french, european, france, cool
46	indelible, screenwriter, obscure, intention
49	directed, written, chris, badly
57	illustrate, express, hot, draw
59	andy, shock, overly, model
62	sensuality, pg-13, correctly, dazzle
64	serial, killer, killing, video
69	routine, howard, lie, gold
75	bad movie, worst, pack, embarrass
79	mom, dad, gag, antic
81	rely, heavily, cliches, charles
103	block, road, desert, area
105	tiny, craft, talented, paint
113	teach, lesson, average, learn
114	particular, crush, year-old, circle
118	ring, lord, false, fate
121	subtitle, english, england, wall
124	quietly, grow, tiresome, yearn
128	equip, stand, transcend, grasp
129	band, rock, magic, remind
141	gorgeous, beauty, opening, stunning
154	gang, crowd, rob, rise

179	personality, billy, majority, studio
193	victim, kidnap, rape, vicious
194	nightmare, horror, dangerous, alive
197	run-in, next, mannerism, well thing
198	big-budget, prove, lovable, cutting
199	hundreds, entire, week, nick
207	revenge, seek, ground, air
211	wake, drink, constantly, spirit
216	knock, door, stranger, trust
222	stream, water, steve, literally
232	brutal, capture, captivate, motive
233	vulnerable, fear, joseph, execute
234	carry, aforementioned, string, pull
236	advise, upbeat, happy, interrupt
237	thousands, dramatic, consist, press
238	rule, hidden, real life, rival
239	veteran, tough, dollar, setup
250	unstoppable, lead, decline, installment
251	hotel, room, lock, loyal
252	alex, alone, suspect, cover
254	effect, special, matthew, sarah
256	proclaim, brief, would-be, casting
262	peer, high, junior, accuse
266	julie, number, borrow, wound
268	edit, strongly, tight, innovative
269	loss, attention, celebrity, italian
270	smoke, frequently, grade, gray
271	curse, stop, corner, guide
272	marry, happily, jane, senior
276	jones, search, scheme, cute
281	ryan, tom, absolute, throw
287	fair, lady, transform, winning
291	confess, highly, outstanding, mesmerize
296	connect, suffer, past, alcoholic

297	present-day, complication, surprise, predict
298	research, evil, mouse, priest
299	real people, attempt, unsuccessful, senator
305	lame, project, stick, remarkable
307	torture, body, naked, skin
308	underwrite, charm, charisma, host
318	kill, assassination, theory, credibility
320	grandmother, mother, clown, middle-aged
322	partially, include, essence, ed
325	letter, poor, heaven, punch
326	considerable, feature, low-key, wayne
342	previously, fill, instantly, preposterous
344	genius, terribly, complete, crash
347	global, old, reservation, political correctness
349	tom hanks, deliciously, cast, isolate
350	internet, miss, passing, bedroom
356	reject, speak, proof, significant
358	great actor, acting, small part, academy award nomination
359	weave, couple, vacation, irritate
360	successfully, pick, possess, humorous
362	sense, handful, valuable, largely
3	suspension, disbelief, suspend
7	lampoon, national, aim
14	robbery, bank, criminal
27	homosexual, gay, bar
29	mpaa, rating, clearly
37	media, news, reporter
41	horrific, frighten, chill
42	wedding, funeral, third
60	employee, boss, mafia
66	sean connery, dr, psychiatrist
67	press kit, favorite, dominate
71	united, state, north
80	central character, affair, attract

90	doctor, medical, patient
93	joy, magical, lovely
100	generous, city, prevent
106	devastate, truth, friendship
108	study, psychological, honest
116	documentary, interview, fire
132	path, cross, mix
136	underrate, receive, effectively
144	fatal, catch, jeff
150	nostalgic, tone, reflect
157	bloody, bad guy, blow
159	cable, tv, douglas
161	blonde, hair, handsome
165	store, shop, station
172	rank, television, hospital
177	professional, hire, assistant
178	emotionally, cold, humanity
182	laughter, hilarious, amusing
185	mature, coming, contain
186	grim, unpleasant, subject
202	frequent, direction, dealer
205	breath, add, massive
208	charmer, fun, blast
217	europe, art, visually
221	stab, full, macho
228	conspiracy, sound, feed
230	ultimately, climactic, mere
240	kick, bill, length
242	circumstance, enter, treatment
243	danger, warn, minor
246	beautifully, test, stuff
247	expose, deserve, variety
253	pure, silly, stupid
259	precious, obvious, reference

263	jesus, michael, tradition
265	texas, drop, boring
267	object, ability, viewing
273	aspiring, close, arrival
274	intriguing, fascinate, notice
275	jay, short, campaign
278	execution, big laugh, david
284	poorly, believable, trouble
285	technique, anderson, paul
286	worthy, agree, earn
289	fan, strictly, responsible
290	inevitable, stage, real-life
301	loud, listen, beat
310	quick, front, dozen
313	sinister, claim, fame
315	wash, open, shut
317	disappoint, shine, bright
321	remains, several, thankfully
328	ward, serious, jungle
329	nominal, hand, virtue
331	wealth, line, parade
332	form, ambition, ancient
333	dignity, screen, clone
334	moviegoer, mind, devote
335	performer, presence, scott
336	bruce, level, amazing
337	touching, happiness, feeling
340	flying, shoot, shakespeare
345	footage, painting, figure
346	safe, count, warm
348	assure, cut, unbelievable
355	develop, necessary, separate
363	longtime, friend, joint
365	helen, propose, meet

366	family, doc, farm
367	meaningful, help, heartfelt
368	fade, real, culminate
369	talk, guest, decent

E.3 Label Propagation 2000 best lift links

Community	Members
36	science fiction, specialised, protagonist, author, masterpiece, great film
10	cia, operative, ww2, military, weapon, lethal, colonel, soldier, troop
23	heartbreaking, deeply, gradually, occasionally, difficult, appearance
17	astronaut, space, sigourney weaver, alien, fiction, science, battlefield
43	sequel, predecessor, indelible, screenwriter, popcorn, superior, summer
32	cgi, technology, futuristic, computer, generate, program, machine, result
0	best actor, best picture, nominate, academy award, nomination, oscar
22	homicide, detective, investigation, investigate, duo, partner, evidence
46	terrorist, political, politician, government, politics, official, destruction
38	bush, president, vice, election, united, state, liberty, staff, nation
20	football, coach, player, team, sport, sports, ball, league, edit, field
30	musical, broadway, crush, year-old, sing, song, band, rock, singer
4	jew, jewish, hitler, nazi, germany, german, risk, flat, success
12	james bond, gadget, fbi, agent, spy, tendency, villain, millions, chase
152	unfortunate, event, theatre, enhance, slow, severely, mostly, showing
21	running, run time, notable, final, consistently, thin, slight, scale
75	bad language, editor, totally, sex, color, brown, magic, guess
92	bully, class, student, high school, teacher, principal, insist, demonstrate
106	likable, standout, admittedly, equip, stand, newcomer, carry
130	unfunny, comedic, timing, spoof, outrageous, waste, humor, crude
18	attorney, trial, court, lawyer, divorce, justice, law
27	sexual innuendo, son jeffrey, complaint, disney, size, adult, age
40	british, britain, indiana, adventure, exotic, london, advance
42	civilisation, inspire, portray, mad, noble, due, sacrifice
52	visual effect, nearly, genre, counterpart, slave, closer, outcome
60	creative, appropriately, delight, garner, hint, likely, witty

66	mature teenager, nudity, frontal, admonish, porn, glad, overact
79	multitude, excellent, depressive, order, colour, star war, bellow
82	sentence, prison, guard, inferior, escape, quiet, barely
97	unknown, era, filmmaker, artist, conventional, similar, elicit
114	premise, intentionally, one-dimensional, sheer, creation, advantage
122	assassin, serial killer, thriller, taut, supernatural, instance, shot
140	ahead, step, wave, palpable, ultimately, climactic, potential
172	particular, circle, initial, new york, hall, shame, fresh
2	caper, heist, crime, petty, plan, crook
16	robbery, bank, criminal, perspective, increasingly, memorable
24	jeffrey, thumb, scare, straight, horrible, extremely
34	set-piece, giant, visual, entertainment, imagination, landscape
72	ship, captain, crew, sea, island, disaster
73	resource, decade, present-day, compensate, connect, citizen
84	blood, gory, gore, flick, bloody, bad guy
89	storyline, proceeding, over-the-top, entertaining, loose, performer
105	artistic, popular, popularity, flawless, universal, important
146	quietly, grow, tiresome, feat, person, yearn
160	realism, bond, tear, central, south, poignant
161	dialog, mainly, stare, la, advise, reply
178	trademark, theme, element, wing, human being, career
13	adrenaline, rush, expectation, aid, disappoint
29	homosexual, gay, sexuality, old teenager, concern
37	favorite part, pg, mild, table, typical
55	dancing, dance, dancer, shoe, folk
56	adam sandler, joke, lazy, raunchy, comedian
62	depth, weight, mainstream, fault, making
64	complicate, further, motion picture, apparent, equal
101	explicit, sexual, seductive, male, female
109	love story, chemistry, romance, blossom, romantic comedy
111	relief, comic, one-liner, throwaway, broad
112	outfit, costume, estate, outlandish, colorful
127	devastate, truth, marvelous, kiss, beach
164	racist, white, racism, print, racial

183	awe-inspire, sequence, alternate, sumptuous, render
187	usa, american, victory, succumb, missile
214	faithful, future, share, dysfunctional, ironically
7	lampoon, national, aim, individual
26	resemblance, bear, snow, overall
28	cartoon, animation, voice, animated
35	pale, comparison, remake, exchange
39	golden age, history, lieutenant, empire
44	media, news, reporter, america
49	retort, hate, rendition, version
57	heroin, drug, addict, habit
59	downright, writer-director, genuine, support
61	historical, epic, tragic, century
65	illustrate, express, moral, hot
69	stunt, explosion, double, thought
77	scary, creepy, attack, frank
81	press kit, favorite, stab, full
96	rely, heavily, cliches, finale
98	thrill, exciting, frame, ride
100	cat, animal, pet, dog
104	context, detail, spare, visually
108	sibling, parent, foster, coming-of-age
117	stereotype, culture, contemporary, explore
121	zoom, camera, pan, angle
128	loving, flashback, childhood, tragedy
134	sad, pathetic, uneven, worse
137	unit, impressive, investigator, material
142	onscreen, focus, dialogue, shift
147	university, college, graduate, professor
148	impact, goal, achieve, fit
166	strip, actual, gain, center
182	unnecessary, subplot, belief, jack
195	composer, quality, score, anymore
211	doom, adaptation, cinematic, patrick

220	hundreds, entire, thousands, dramatic
243	unexpected, latter, enter, treatment
3	suspension, disbelief, suspend
5	expression, facial, arrive
41	thick, accent, southern
45	lead role, leading, well friend
47	horrific, frighten, chill
48	wedding, funeral, marry
50	french, european, france
58	portrayal, accurate, consequence
63	authority, convincing, source
68	employee, boss, mafia
74	industrial, period, reality
76	serial, killer, killing
78	sean connery, dr, psychiatrist
87	west, east, western
90	bad movie, worst, pack
91	sympathy, root, sympathetic
94	mom, dad, antic
103	gun, bullet, shooting
110	doctor, medical, patient
113	joy, magical, lovely
118	sight, gag, journey
119	amateurish, walk, fail
123	block, road, desert
125	tiny, craft, carefully
129	study, psychological, honest
131	disturb, vision, freedom
133	ideal, match, known
135	digit, single, awe
141	ring, lord, false
143	subtitle, english, england
144	nominal, today, execution
149	unhappy, marriage, married

151	undercover, police, arrest
156	well film, brilliant, depiction
157	satire, public, wise
158	effective, underrate, receive
162	client, company, executive
165	fatal, catch, jeff
167	yes, amaze, make-up
169	comrade, lack, sorely
171	likewise, refuse, smart
173	homage, pay, price
174	competition, perform, young man
176	gang, crowd, rob
177	embrace, serve, creator
184	locale, fine, stirring
185	palace, view, shatter
186	store, shop, station
188	tune, pop, accept
189	infinitely, far, assured
190	furious, fast, bound
194	climax, unlikely, outside
196	flow, fantasy, lonely
199	emotionally, cold, humanity
200	personality, billy, majority
201	admirable, hit, mixed
202	laughter, hilarious, amusing
208	gangster, heavy, business
210	theatrical, release, faith
212	practically, major, discover
213	victim, kidnap, rape
216	entry, fellow, inhabit
221	tired, matter, co-star
222	annoy, present, experiment
224	genius, personal, deny
225	busy, wear, hat

229	violent, filmmaking, cliched
231	natural, empty, conversation
239	rent, party, core
240	delightful, dress, screening
244	truck, drive, crazy
247	condition, rich, heat
248	brutal, capture, captivate
251	upbeat, happy, interrupt
256	designer, production, design

Community	Members
27	running, run time, minute run time, heartbreaking, deeply, gradually, giant
34	specialised, protagonist, pale, comparison, civilisation, author, masterpiece
29	bad part, worst, jeffrey, thumb, bear, resemblance, homosexual, gay, mpaa
20	cutting, cut room floor, supporting, support role, support character
28	football, coach, open scene, opening, player, team, sport, joy, magical
32	ww2, military, weapon, lethal, soldier, colonel, bush, president, troop
13	heist, caper, law, enforcement, robbery, bank, attorney, trial, homicide
26	work today, working, space, astronaut, sigourney weaver, alien, fiction
18	james bond, gadget, sexual innuendo, son jeffrey, cartoon, animation
6	jesus, christ, taxi, driver, special effects, special, effect, heroin, drug
17	best actor, best picture, nominate, niro, de, academy award, nomination
19	politically, correct, musical, broadway, dancing, dance, dancer, loving
24	multitude, poke, fun, present-day, all-powerful, order, depressive, colour
31	national, lampoon, adam sandler, joke, aim, mom, dad, gag, comedic
15	placement, product, lead role, leading, wedding, funeral, tale, fairy
9	expression, facial, adrenaline, rush, thick, accent, southern, zoom, camera
33	silence, lamb, gesture, simple, effective, underrate, receive, flawless
16	jones, indiana, plane, flight, british, britain, adventure, french
11	jew, jewish, nazi, hitler, dinner, andre, germany, german, worker, factory
25	bully, class, student, high school, teacher, teach, lesson, increasingly
21	cia, operative, sequel, predecessor, indelible, screenwriter, popcorn
22	disbelief, suspension, suspend, written, directed, pierce, able, poorly
8	denzel, washington, d.c., fortunate, place, down-to-earth, point of view